RM03/02

DISKLESS PART 2

CZRMKA0

AH-E377A-MC

COPYRIGHT © 1978

FICHE 1 OF 1

JUL 1978

digital

MADE IN USA

```
                                .REM
```

IDENTIFICATION

PRODUCT CODE:	AC-E376A-MC
PRODUCT NAME:	CZRMKA0 RM03/RM02 DISKLESS DIAGNOSTIC, PART II
DATE CREATED:	1-MARCH-78
MAINTAINER:	DIAGNOSTIC GROUP
AUTHOR:	DOUG RIIKONEN

THE INFORMATION IN THIS DOCUMENT IS SUBJECT TO CHANGE WITHOUT NOTICE AND SHOULD NOT BE CONSTRUED AS A COMMITMENT BY DIGITAL EQUIPMENT CORPORATION. DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR ANY ERRORS THAT MAY APPEAR IN THIS MANUAL.

THE SOFTWARE DESCRIBED IN THIS DOCUMENT IS FURNISHED TO THE PURSHASER UNDER A LICENSE FOR USE ON A SINGLE COMPUTER SYSTEM AND CAN BE COPIED (WITH INCLUSION OF DIGITAL'S COPYRIGHT NOTICE) ONLY FOR USE IN SUCH SYSTEM, EXCEPT AS MAY OTHERWISE BE PROVIDED IN WRITING BY DIGITAL.

DIGITAL EQUIPMENT CORPORATION ASSUMES NO RESPONSIBILITY FOR THE USE OR RELIABILITY OF ITS SOFTWARE ON EQUIPMENT THAT IS NOT SUPPLIED BY DIGITAL.

COPYRIGHT (C) 1978, DIGITAL EQUIPMENT CORPORATION

CONTENTS

1. INTRODUCTION

    1. ABSTRACT

    2. UNIT UNDER TEST

2. OPERATING REQUIREMENTS

    1. HARDWARE REQUIREMENTS

    2. MEDIA REQUIREMENTS

    3. PREREQUISITE DIAGNOSTIC PROGRAMS

3. OPERATING PROCEDURE

    1. LOADING

    2. SWITCH OPTIONS

    3. STARTING

4. OPERATOR INTERFACE

    1. PROGRAM ID

    2. PROGRESS REPORTS

    3. PERFORMANCE REPORTS

    4. PROGRAM HALTS

    5. ERROR REPORTS

5. ENVIRONMENTAL SUPPORT

    1. PROCESSOR COMPATIBILITY

    2. DUAL PORT CONFIGURATIONS

    3. MEMORY PARITY HARDWARE

    4. MEMORY MANAGEMENT HARDWARE

    5. ACT,APT COMPATIBILITY

6. XXDP COMPATIBILITY

7. OPERATING SYSTEM COMPATIBILITY

6. TEST DESCRIPTION

1.0 INTRODUCTION

1.1 ABSTRACT

THE RM03 DISKLESS DIAGNOSTIC IS A STAND ALONE PROGRAM WHICH USES FUNCTIONAL AND DIAGNOSTIC MEANS TO VERIFY THE OPERABILITY OF THE RM03 DISK SUBSYSTEM EXCLUDING AND INDEPENDENTLY OF THE STORAGE MODULE DRIVE. IN PARTICULAR, THE PROGRAM SERVES THE FOLLOWING PURPOSES:

TO DETECT ERRORS AND FAULTS IN THE RM03 MASSBUS ADAPTER;

TO RESOLVE HARDWARE FAILURES IN RM03 TO A FIELD REPLACEABLE MODULE OR MODULES.

1.2 UNIT UNDER TEST

THE UNIT UNDER TEST (UUT) IS THE RM03 DISK SUBSYSTEM, EXCLUDING THE STORAGE MODULE DISK DRIVE AND THE MASSBUS CONTROLLER.

2.0 OPERATING REQUIREMENTS

2.1 HARDWARE REQUIREMENTS

THE FOLLOWING MINIMUM HARDWARE CONFIGURATION, ASSUMED TO BE OPERATIONAL, IS REQUIRED TO LOAD AND EXECUTE THE RM03 DISKLESS DIAGNOSTIC:

PDP-11 PROCESSOR
16 K MEMORY
KW11-L OR LW11-P CLOCK
PROGRAM LOADING DEVICE
TERMINAL
RH CONTROLLER
UNIT UNDER TEST,
WHERE THE UNIT UNDER TEST CONSISTS OF ONE TO EIGHT RM03 ADAPTERS.

2.2 MEDIA REQUIREMENTS

NONE

2.3 PREREQUISITE DIAGNOSTIC PROGRAMS

CZRMJ- ,RM03/02 DISKLESS DIAGNOSTIC PART 1

3.0 OPERATING PROCEDURE

3.1 LOADING

THE PROGRAM MAY BE LOADED BY EITHER PAPER TAPE, USING THE STANDARD PAPER TAPE LOADING PROCEDURE, OR XXDP MEDIA, USING THE APPROPRIATE LOADING DEVICE.

3.2 SWITCH OPTIONS

THE FOLLOWING SWITCH OPTIONS ARE INVOKED WHEN THE APPROPRIATE SWITCH IS ON.

SW15	HALT ON ERROR
SW14	LOOP ON TEST (CURRENTLY BEING EXECUTED)
SW13	INHIBIT ERROR TYPEOUTS
SW12	UNUSED
SW11	INHIBIT TEST ITERATIONS
SW10	BELL ON ERROR
SW09	LOOP ON ERROR
SW08	LOOP ON TEST IN SW07-00

THE LOW ORDER 8 SWITCHES ARE USED IN CONJUNCTION WITH SW08 TO SPECIFY THE OCTAL NUMBER OF THE TEST WHICH THE PROGRAM WILL LOOP ON.

3.3 STARTING

THE PROGRAM STARTS AT LOCATION 200 WHICH USES DEFAULT VALUES OF PARAMETERS AND PROVIDES MAXIMUM TESTING WITH THE SWITCH REGISTER EQUAL TO ZERO.

4.0 OPERATOR INTERFACE

4.1 PROGRAM ID

THE PROGRAM TYPES ITS NAME AND MAINDEC NUMBER THE FIRST TIME IT IS STARTED AFTER BEING LOADED.

4.2 PROGRESS REPORTS

AN END OF PASS REPORT OCCURRS EACH TIME THE PROGRAM IS EXECUTED FOR ALL ADAPTERS IN THE TEST QUE. THE END OF PASS REPORT INCLUDES A MESSAGE AND AN ERROR SUMMARY.

4.3 PERFORMANCE REPORT

NO PERFORMANCE REPORTS ARE GIVEN DURING THE EXECUTION OF THE PROGRAM.

4.4 PROGRAM HALTS

THERE ARE NO SCHEDULED HALTS DURING THE EXECUTION OF THE PROGRAM. PROCESSOR HALTS ARE DUE TO THE TRAP CATCHER.

4.5 ERROR REPORTS

THE RM03 DISKLESS DIAGNOSTIC PROVIDES COMPREHENSIVE ERROR REPORTS INTENDED TO (1) AID IN FAULT RESOLUTION AND (2) MINIMIZE REFERENCES TO PROGRAM LISTINGS.

THE FIRST LINE OF THE ERROR REPORT CONTAINS THE NUMBER OF THE UNIT BEING TESTED, THE TEST NUMBER, THE ERROR NUMBER AND THE VALUE OF THE PROGRAM COUNTER WHERE THE ERROR WAS CALLED. THIS LINE IS FOLLOWED BY THE ERROR MESSAGE: SEVERAL LINES OF TEXT WHICH GIVE A COMPREHENSIVE DESCRIPTION OF THE ERROR, AND A LIST OF FAILING MODULES IN ORDER OF DECREASING PROBABILITY. THE ERROR MESSAGE IS NORMALLY FOLLOWED BY ONE OR MORE PAIRS OF LINES CONTAINING DATA HEADERS AND DATA PERTININENT TO THE ERROR, INCLUDING EXPECTED AND ACTUAL TEST

```
   265                                  RESULTS.
```

5.0 ENVIRONMENTAL SUPPORT

5.1 PROCESSOR COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC IS EXECUTABLE ON ANY PDP-11 PROCESSOR, PROVIDING PREVIOUSLY MENTIONED HARDWARE REQUIREMENTS ARE MET.

5.2 DUAL PORT CONFIGURATIONS

THE RM03 DISKLESS DIAGNOSTIC IS NOT EXECUTABLE ON RM03 SUBSYSTEMS HAVING THE DUAL PORT OPTION UNLESS THE DUAL PORT SWITCH IS SET TO THE APPROPRIATE PORT (A OR B) AND NOT TO THE PROGRAMMABLE POSITION (A/B).

5.3 MEMORY PARITY HARDWARE

MEMORY PARITY HARDWARE WILL NOT BE USED DURING THE EXECUTION OF THE RM03 DISKLESS DIAGNGOSTIC.

5.4 MEMORY MANAGEMENT HARDWARE

MEMORY MANAGEMENT HARDWARE WILL NOT BE USED DURING THE RM03 DISKLESS DIAGNOSTIC.

5.5 ACT11, APT11 COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC PROGRAM IS COMPATIBLE WITH ACT11 AND APT11 IN BOTH DUMP AND AUTOMATIC MODES. FURTHER, THE PROGRAM WILL EXECUTE A QUICK PASS DURING THE FIRST PASS IN SUPPORT OF QUICK VERIFY MODE.

5.6 XXDP COMPATIBILITY

THE RM03 DISKLESS DIAGNOSTIC PROGRAM IS COMPATIBLE WITH XXDP IN DUMP AND CHAIN MODES.

5.7 OPERATING SYSTEM COMPATIBILITY

THE PROGRAM IS NOT REQUIRED TO BE COMPATIBLE WITH ANY OPERATING SYSTEM.

6.0 TEST DESCRIPTION

THE PROGRAM IS DESIGNED IN A BOTTOM UP MANNER SUCH THAT EACH TEST GENERALLY USES A MORE COMPLEX SUBSET OF HARDWARE THAN THE PREVIOUS TEST.

MODULE CALLOUT IS PREDICATED ON THE ASSUMPTION THAT EARLIER TESTS HAVE BEEN COMPLETED WITHOUT ERROR AND THAT ERRORS ARE DUE TO SINGLE, NONTRANSIENT HARDWARE FAILURES.

THE "RM03 DISKLES DIAGNOSTIC" CAN BE EXECUTED USING AN RH70 OR AN RH11 MASSBUS CONTROLLER.

UNLESS SPECIFIED BY THE OPERATOR OR BY THE ENVIRONMENT TABLE THE TEST IS REPEATED FOR EACH POSSIBLE DEVICE STARTING WITH DEVICE 0.

THE MODULES WHICH MAY BE CALLED OUT DURING THE EXECUTION OF THE TEST ARE AS FOLLOWS:

IF
CS
DS
MASSBUS MODULE

THE RADIAL MODULE (RD) IS NOT TESTED BY THIS PROGRAM.

TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE RM03 CAN COMPLETE A REGISTER TRANSFER ON THE MASSBUS, AND, IN PARTICULAR, TO VERIFY THAT "TRANSFER" IS NOT STUCK IN AN INACTIVE STATE.

PROCEDURE:

THE PROGRAM WRITES AND READS REMOTE REGISTERS FOR THE SELECTED DEVICE. REGISTER CONTENTS AND PARITY ERRORS ARE IGNORED, AND THE TEST FAILS IF A "NONEXISTENT DEVICE ERROR" OR BUS TIMEOUT OCCURRS FOR EVERY REGISTER ACCESS. IF THE TEST FAILS THE PROGRAM JUMPS TO THE END OF PASS HANDLER WHICH SELECTS THE NEXT DEVICE TO BE TESTED.

PROBABLE FAULT:

THE TEST FAILS IF THE SELECTED DEVICE IS NONEXISTENT OR IS SWITCHED TO THE PROGRAMMABLE POSITION OR TO THE ALTERNATE PORT. THE FOLLOWING FAULTS ARE APPLICABLE ONLY WHEN THE DEVICE IS PRESENT AND IS SWITCHED TO THE APPROPRIATE PORT.

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

3. CS MODULE

CTOD TEST

PURPOSE:

TO VERIFY THAT DATA CAN BE TRANSFERRED TO AND FROM THE RM03 USING THE CONTROL BUS AND, IN PARTICULAR, TO VERIFY THAT "CONTROLLER TO DEVICE" HAS NOT FAILED.

PROCEDURE:

THE TEST WRITES ONES IN REMOTE REGISTERS THEN READS EACH REGISTER WHICH WILL WRITE ZEROS IN THE REGISTER IF "IF3 CTOD HOLD H" IS STUCK AT ONE. THE TEST THEN READS AS MANY REMOTE REGISTERS AS ARE NECESSARY TO OBTAIN ONE OR MORE ONE BITS.

PROBABLE FAULT:

1. IF MODULE

2. ASYNCHRONOUS MASSBUS MODULE

MASSBUS INITIALIZE TEST

PURPOSE:

TO VERIFY THAT THE MASSBUS ADAPTER IS BEING INITIALIZED BY THE MASS BUS.

PROCEDURE:

USING CONTROLLER CLEAR TO INITIALIZE THE SELECTED UNIT, THIS TEST THEN READS MASSBUS ADAPTER REGISTERS TO VERIFY THAT AT LEAST ONE BIT IS CLEARED. MASSBUS ADAPTER REGISTERS ARE PRESET TO A NON ZERO VALUE PRIOR TO CONTROLLER CLEAR.

PROBABLE FAULT:

1. ASYNCHRONOUS MASSBUS MODULE

2. IF MODULE

3. CS MODULE

CLEAR STUCK ACTIVE TEST

PURPOSE:

TO VERIFY THAT "MBA CLR L" ON THE CS MODULE IS NOT STUCK IN AN ACTIVE STATE.

PROCEDURE:

CONTROLLER CLEAR IS USED TO INITIALIZE THE SELECTED UNIT, AFTER WHICH 1'S ARE WRITTEN IN ERROR REGISTERS 1 AND 2 AND MAINTENANCE REGISTER 1. IF ANY 1 BITS CAN BE READ BACK THE TEST IS OK, ELSE, "MBA CLR L" IS PROBABLY STUCK ACTIVE.

PROBABLE FAULT:

1. CS MODULE

2. IF MODULE

3. ASYNCHRONOUS MASSBUS MODULE

TRISTATE TRANSFER TEST

PURPOSE:

TO VERIFY THAT THE PATH TO AND FROM THE MASSBUS ADAPTER TRI-STATE REGISTER BUS IS NOT STUCK AT ONE OR ZERO AND THAT EACH BIT POSITION IS INDEPENDENT.

PROCEDURE:

THIS TEST PRESETS MASSBUS ADAPTER REGISTERS TO A NONZERO VALUE, THEN, ASSUMING THE REGISTERS ARE PRESET, IT CLEARS THEM USING A MOVE INSTRUCTION. THE TEST THEN READS AS MANY REGISTERS AS IS NECESSARY TO OBTAIN ONE OR MORE ZEROS FROM EACH BIT POSITION.

```
   488
```

THE TEST CLEARS MASSBUS ADAPTER REGISTERS, THEN, ASSUMING THE REGISTERS ARE CLEARED, IT LOADS THEM WITH ONES AND READS AS MANY REGISTERS AS IS NECESSARY TO OBTAIN ONE OR MORE ONE BITS IN EACH BIT POSITION.

FINALLY, THE TEST WRITES A SINGLE ONE BIT PATTERN IN BIT 0 OF SELECTED REMOTE REGISTERS AND VERIFIES THAT THE PATTERN CAN BE READ BACK. THE ONE BIT IS SHIFTED AND THE TEST REPEATED FOR ALL BIT POSITIONS.

PROBABLE FAULT:

1. ASYNCHRONOUS MASSBUS MODULE
2. IF MODULE
3. CS MODULE
4. DS MODULE

REGISTER SELECT TEST

PURPOSE:

TO VERIFY THAT THE REGISTER SELECT LINES ARE NOT IN A STUCK POSITION.

PROCEDURE:

EACH REGISTER SELECT LINE IS TESTED BY WRITING ZEROS IN THOSE DEVICE REGISTERS FOR WHICH THE LINE MUST BE ZERO, THEN WRITING ONES IN THOSE DEVICE REGISTERS FOR WHICH THE LINE MUST BE ONE. THE ZERO REGISTER IS READ BACK AND IF THE SELECT LINE IS STUCK AT ZERO, THE ZERO REGISTER WILL CONTAIN ONES. THE PROCESS IS REPEATED TO DETECT A STUCK AT ONE FAULT, EXCEPT IN THIS CASE, THE ONES REGISTER IS WRITTEN FIRST.

REGISTER SELECT LINES 1, 2, 4 AND 8 ARE TESTED IN THIS MANNER; SELECT LINE 16 IS EXPLICITLY TESTED IN THE "ILR TEST".

PROBABLE FAULT:

1. IF MODULE
2. ASYNCHRONOUS MASSBUS MODULE

```
                                DRIVE TYPE TEST

                                        PURPOSE:
                                                TO TEST THE "DRIVE TYPE" REGISTER, RMDT.

                                        PROCEDURE:

                                                THE PROGRAM READS RMDT AND VERIFIES THAT THE RESULT
                                        CORRESPONDS TO A SINGLE PROT OR DUAL PROT RM03 DRIVE

                                        PROBABLE FAULT:

                                        1. IF MOULE

                                DEVICE AVAILABLE TEST

                                        PRUPOSE:
                                        TO VERIFY THAT DEVICE AVAILABLE STATUS IS SET.

                                        PROCEDURE:

                                        THE PROGRAM TESTS "DVA",BIT 11 OF RMCS1.

                                        PROBABLE FAULT:

                                        1. IF MODULE




                        SEARCH TIMEOUT TEST

                                PURPOSE:

                                     TO VERIFY THAT THE SEARCH TIMEOUT ONE SHOT SETS "OPI",
                                EXCEPT WHEN "SEARCH TO DISABLE" IS ACTIVE.

                                PROCEDURE:

                                     WITH SEARCH TIMEOUT DISABLED, THE TEST EXECUTES A DATA
                                COMMAND TO THE POINT WHERE "P ENABLE SEARCH" IS ASSERTED.
                                AFTER WAITING A SUFFICIENT PERIOD AND VERIFYING THAT OPI
                                IS NOT SET, THE TEST ENABLES SEARCH TIMEOUT AND VERIFIES
                                THAT OPI SETS.

                                PROBABLE FAULT:

                                1.  CS MODULE

                                        NOTE:   IT IS ASSUMED THAT THE "SET OPI TEST" HAS
                                                ALREADY PASSED, THUS MAKING THE IF MODULE
                                                AN IMPROBABLE FAULT.
```

```
                                SET DTE TEST

                                        PURPOSE:

                                             IN ADDITION TO VERIFYING THAT "DRIVE TIMING ERROR" CAN
                                        BE SET BY THE CS MODULE, THIS TEST ALSO VERIFIES

                                        *    THAT "MAINTENANCE SECTOR COMPARE" IS NOT STUCK AT ONE
                                             OR ZERO.

                                        *    THAT "ENABLE SEARCH" IS NOT STUCK AT ONE OR ZERO.

                                        PROCEDURE:

                                             (1)  INITIALIZE AND VERIFY THAT "DTE" IS RESET, THEN
                                        SET MAINTENANCE INDEX PULSE AND VERIFY THAT DTE IS STILL
                                        RESET.  THIS TEST WILL INSURE THAT THE SECTOR COMPARE FLOP
                                        IS NOT STUCK AT ONE.

                                             (2)  EXECUTE A DATA COMMAND IN MAINTENANCE MODE TO
                                        THE POINT WHERE SEARCH IS ENABLED, I.E., P EN SEARCH H IS
                                        IS ACTIVE.  SET AND RESET THE SECTOR PULSE TO SET "CS3 EN
                                        SEARCH" FLOP, AND CLOCK "CSS SECTOR COMPARE" FLOP WHICH
                                        SHOULD NOT SET.  SET SECTOR PULSE AND VERIFY THAT DTE IS
                                        RESET.  THIS TEST FAILS IF J INPUT TO SECTOR COMPARE FLOP
                                        IS STUCK AT ONE.

                                             REPEAT, BUT SET MAINTENANCE SECTOR COMPARE AND VERIFY
                                        THAT DTE SETS.  THIS TEST FAILS IF MAINTENANCE SECTOR
                                        COMPARE IS STUCK AT ZERO.

                                        PROBABLE FAULT:

                                        1.  CS MODULE

                                        2.  DS MODULE




                                FORMAT CHANGE TEST

                                        PURPOSE:

                                             TO VERIFY THAT A CHANGE IN FORMAT INHIBITS SEARCH
                                        ENABLE UNTIL THE NEXT INDEX PULSE.

                                        PROCEDURE:

                                             THE TEST WILL USE "DTE" FOR VISIBILITY OF "CS3 EN
```

```
                                    SEARCH H".

                                         THE FOLLOWING STEPS ARE EXECUTED:

                                         (1)  INITIALIZE AND SET THE FORMAT TO 18 BIT MODE
                                    TO SET "CS3 FMT CHANGE" FLOP.

                                         (2)  EXECUTE A DATA COMMAND TO THE POINT WHERE SEARCH
                                    IS ENABLED BY THE SEQUENCER.

                                         (3)  SET "MAINTENANCE SECTOR COMPARE", THEN SET
                                    "MAINTENANCE SECTOR PULSE" TO CLOCK "CS3 EN SEARCH" FLOP
                                    WHICH SHOULD NOT SET BECAUSE OF THE FORMAT CHANGE.

                                         (4)  RESET SECTOR PULSE TO CLOCK "CS5 SECTOR COMPARE"
                                    FLOP WHICH WILL NOT SET IF "CS3 EN SEARCH H" IS INACTIVE.

                                         (5)  SET SECTOR PULSE AND VERIFY THAT DRIVE TIMING
                                    ERROR IS RESET.

                                         (6)  SET AND RESET INDEX PULSE TOCLEAR THE FORMAT
                                    CHANGE FLOP.

                                         (7)  SET AND RESET SECTOR PULSE TO SET "CS3 EN SEARCH"
                                    FLOP AND "CS5 SECTOR COMPARE" FLOP.

                                         (8)  SET SECTOR COMPARE AND VERIFY THAT DTE IS SET.

                                         REPEAT THE TEST WITH A FORMAT CHANGE FROM 18 BIT MODE
                                    TO 16 BIT MODE.

                                    PROBABLE FAULT:

                                    1.  CS MODULE




                               PROM STROBE TEST

                                    PURPOSE:

                                         TO VERIFY THAT WORD CLOCK AND PROM STROBE CAN BE MAN-
                                    IPULATED IN MAINTENANCE MODE.

                                    PROCEDURE:

                                         INITIALIZE AND SET 16 BIT MODE, THEN SEQUENCE THE
                                    MAINTENANCE CLOCK UNTIL PROM STROBE SETS.  ISSUE -- MORE
                                    MAINTENANCE CLOCK PULSES AND VERIFY THAT PROM STROBE RESETS.

                                    PROBABLE FAULT:

                                    1.  CS MODULE
```

2. DS MODULE

SYNC WORD COUNT INHIBIT TEST

PURPOSE:

TO VERIFY THE FOLLOWING DURING READ COMMAND:

* THAT "CS4 P LFS" (LOOKING FOR SYNC) GOES ACTIVE.

* THAT "LOOKING FOR SYNC" INHIBITS THE WORD COUNT

PROCEDURE:

A READ COMMAND IS SETUP AND EXECUTED TO THE POINT WHERE "LOOKING FOR SYNC" SHOULD BE ACTIVE, WITH THE PROGRAM VERIFYING THE TRANSITION OF THE SIGNAL. THE PROGRAM THEN SUPPLIES A SERIES OF BIT CLOCKS AND VERIFIES THAT "PROM STROBE" NEVER GOES ACTIVE.

PROBABLE FAULT:

1. CS MODULE IF LFS FAILT,

2. DS MODULE IF PROM STROBE FAILS.

SYNC DETECTION TEST

PURPOSE:

TO VERIFY THAT THE APPROPRIATE SYNC PATTERN IS RECOGNIZED.

PROCEDURE:

THE TEST EXECUTES A READ COMMAND IN MAINTENANCE MODE, USING BIT 10 OF THE MAINTENANCE REGISTER (RMMR1) TO SIMULATE THE SYNC BIT STREAM, AND USING PROM STROBE TO DETERMINE IF THE SYNC PATTERN HAS BEEN DETECTED.

THE SYNC PATTERN IS 00011001, WITH THE LEFT MOST BIT REPRESENTING THE LAST BIT OF THE STREAM.

PROBABLE FAULT:

1. DS MODULE

2. CS MODULE

```
                                ABORT SYNC DETECTION TEST

                                        PURPOSE:

                                             TO VERIFY THAT "WORD COUNT INHIBIT" IS RESET IF A
                                        "DRIVE TIMING ERROR" OCCURS DURING SYNC DETECTION.

                                        PROCEDURE:

                                             A READ COMMAND IS INITIATED IN MAINTENANCE MODE.  WHEN
                                        "LOOKING FOR SYNC" GOES ACTIVE, THE TEST FORCES A DRIVE
                                        TIMING ERROR AND USES PROM STROBE TO VERIFY THAT "WORD
                                        COUNT INHIBIT" HAS BEEN RESET.

                                        PROBABLE FAULT:

                                        1.  DS MODULE



                                SYNC GENERATION TEST

                                        PURPOSE:

                                             TO VERIFY THAT THE APPROPRIATE SYNC PATTERN IS
                                        GENERATED DURING A FORMAT OPERATION.

                                        PROCEDURE:

                                             THE TEST EXECUTES A WRITE HEADER AND DATA COMMAND
                                        IN MAINTENANCE MODE, AND VERIFIES THAT THE CORRECT SYNC
                                        PATTERN IS GENERATED BY MONITORING WRITE DATA AT THE MAIN-
                                        TENANCE REGISTER (RMMR1).

                                        PROBABLE FAULT:

                                        1.  DS MODULE



                                WRITE HEADER TEST

                                        PRUPOSE:

                                                TO TEST THE OPERATION OF (1) THE DATA BUFFER AND SHIFT
                                        REGISTER AS WELL AS (2) THE ECC GENERATION DURING WRITE.

                                        PROCEDURE:
```

A WRITE HEADER AND DATA COMMAND IS EXECUTED IN MAINTENANCE MODE. THE TEST VERIFIES HEADER WORDS ONE AND TWO AND THE CRC CHARACTER USING THE WRITE DATA BIT OF THE MAINTENANCE REGISTER.

THE TEST USES CYLINDER 822, TRACK 4, SECTOR 31 AND 16 BIT FORMAT, WHICH CORRESPONS TO THE FOLLOWING HEADER:

WORD 1 - 1101001100110110

WORD 2 - 0000010000011111

PROBABLE FAULT:

DS MODULE OR MASSBUS MODULE

HEADER COMPARE TEST

PURPOSE:

TO CHECK THE OPERATION OF (1) THE SHIFT REGISTER AND DATA BUFFER AS WELL AS THE (2) CRC GENERATOR DURING READ.

PROCEDURE:

THE TEST EXECUTES A READ HEADER AND DATA COMMAND IN MAINTENANCE MODE USING BIT 10 OF THE MAINTENANCE REGISTER (RMMR1) TO SIMULATE THE DATA BITS FOR THE FIRST AND SECOND HEADER WORDS AS WELL AS THE CRC CHARACTER. THE CRC CHARACTER IS IS FAULTED, AND THE TEST VERIFIES THAT A CRC ERROR IS DETECTED. ADDITIONALLY, THE TEST VERIFIES THAT HEADER WORDS ONE AND TWO ARE CORRECTLY TRANSFERD TO MEMORY.
THE TEST USES THE SAME HEADER AS IN THE PREVIOUS TEST EXCEPT THAT BIT 15 IS INVERTED.

PROBABLE FAULT:
DS OR IF MODULE IF CRC ERROR NOT DETECTED;
DS OR MASSBUS MODULE IF DATA INCORRECT

ECC GENERATION TEST

PURPOSE:

TO CHECK ECC OPERATION DURING WRITE.

PROCEDURE:

A WRITE DATA COMMAND IS EXECUTED IN MAINTENANCE MODE. ALL ONES DATA FIELD IS USED, AND THE TEST VERIFIES THE ECC CHARACTER VIA THE WRITE DATA BIT OF THE MAINTENANCE REGISTER. THE DATA FIELD IS NOT VERIFIED.

```
880
881                                     PROBABLE FAULT:
882                                     DS MODULE
883
884                             ECC     DETECTION TEST
885
886                                     PURPOSE:
887
888                                             TO CHECK THE ECC GENERATION DURING READ.
889
890                                     PROCEDURE:
891
892                                             THE TEST EXECUTES A READ DATA COMMAND IN MAINTENANCE
893                                     MODE, AN ALL ONES DATA FIELD IS USED, HOWEVER THE LAST DATA
894                                     WORD IS FAULTED.
895
896                                             THE TEST VERIFIES (1) THAT AN ECC ERROR IS DETECTED
897                                     AND THAT (2) THE POSITION AND PATTERN REGISTERS ARE VALID.
898
899                                     PROBABLE FAULT:
900
901                                     DS MODULE
902
903
```

```
                                                    I  2
                                ;PROGRAM REVISION #001

                                .TITLE  CZRMKA0 RM03/2 DSKLS PRT 2
                                ;*COPYRIGHT (C) 1977
                                ;*DIGITAL EQUIPMENT CORP.
                                ;*MAYNARD, MASS. 01754
                                ;*
                                ;*PROGRAM BY DOUG RIIKONEN
                                ;*
                                ;*THIS PROGRAM WAS ASSEMBLED USING THE PDP-11 MAINDEC SYSMAC
                                ;*PACKAGE (MAINDEC-11-DZQAC-C3), JAN 19, 1977.
                                ;*
000001                          $TN=1
                                .SBTTL  OPERATIONAL SWITCH SETTINGS
                                ;*
                                ;*      SWITCH                  USE
                                ;*      ------          --------------------
                                ;*        15            HALT ON ERROR
                                ;*        14            LOOP ON TEST
                                ;*        13            INHIBIT ERROR TYPEOUTS
                                ;*        11            INHIBIT ITERATIONS
                                ;*        10            BELL ON ERROR
                                ;*         9            LOOP ON ERROR
                                ;*         8            LOOP ON TEST IN SWR<7:0>
                                ;*         7            TN128
                                ;*         6            TN64
                                ;*         5            TN32
                                ;*         4            TN16
                                ;*         3            TN8
                                ;*         2            TN4
                                ;*         1            TN2
                                ;*         0            TN1
                                .SBTTL  BASIC DEFINITIONS

                                ;*INITIAL ADDRESS OF THE STACK POINTER *** 1100 ***
001100                          STACK=  1100
                                .EQUIV  EMT,ERROR       ;;BASIC DEFINITION OF ERROR CALL
                                .EQUIV  IOT,SCOPE       ;;BASIC DEFINITION OF SCOPE CALL

                                ;*MISCELLANEOUS DEFINITIONS
000011                          HT=     11              ;;CODE FOR HORIZONTAL TAB
000012                          LF=     12              ;;CODE FOR LINE FEED
000015                          CR=     15              ;;CODE FOR CARRIAGE RETURN
000200                          CRLF=   200             ;;CODE FOR CARRIAGE RETURN-LINE FEED
177776                          PS=     177776          ;;PROCESSOR STATUS WORD
                                .EQUIV  PS,PSW
177774                          STKLMT= 177774          ;;STACK LIMIT REGISTER
177772                          PIRQ=   177772          ;;PROGRAM INTERRUPT REQUEST REGISTER
177570                          DSWR=   177570          ;;HARDWARE SWITCH REGISTER
177570                          DDISP=  177570          ;;HARDWARE DISPLAY REGISTER

                                ;*GENERAL PURPOSE REGISTER DEFINITIONS
000000                          R0=     %0              ;;GENERAL REGISTER
000001                          R1=     %1              ;;GENERAL REGISTER
000002                          R2=     %2              ;;GENERAL REGISTER
000003                          R3=     %3              ;;GENERAL REGISTER
```

```
                                R4=     %4              ;;GENERAL REGISTER
                                R5=     %5              ;;GENERAL REGISTER
                                R6=     %6              ;;GENERAL REGISTER
                                R7=     %7              ;;GENERAL REGISTER
                                SP=     %6              ;;STACK POINTER
                                PC=     %7              ;;PROGRAM COUNTER

                                ;*PRIORITY LEVEL DEFINITIONS
                                PR0=    0               ;;PRIORITY LEVEL 0
                                PR1=    40              ;;PRIORITY LEVEL 1
                                PR2=    100             ;;PRIORITY LEVEL 2
                                PR3=    140             ;;PRIORITY LEVEL 3
                                PR4=    200             ;;PRIORITY LEVEL 4
                                PR5=    240             ;;PRIORITY LEVEL 5
                                PR6=    300             ;;PRIORITY LEVEL 6
                                PR7=    340             ;;PRIORITY LEVEL 7

                                ;*"SWITCH REGISTER" SWITCH DEFINITIONS
                                SW15=   100000
                                SW14=   40000
                                SW13=   20000
                                SW12=   10000
                                SW11=   4000
                                SW10=   2000
                                SW09=   1000
                                SW08=   400
                                SW07=   200
                                SW06=   100
                                SW05=   40
                                SW04=   20
                                SW03=   10
                                SW02=   4
                                SW01=   2
                                SW00=   1
                                .EQUIV  SW09,SW9
                                .EQUIV  SW08,SW8
                                .EQUIV  SW07,SW7
                                .EQUIV  SW06,SW6
                                .EQUIV  SW05,SW5
                                .EQUIV  SW04,SW4
                                .EQUIV  SW03,SW3
                                .EQUIV  SW02,SW2
                                .EQUIV  SW01,SW1
                                .EQUIV  SW00,SW0

                                ;*DATA BIT DEFINITIONS (BIT00 TO BIT15)
                                BIT15=  100000
                                BIT14=  40000
                                BIT13=  20000
                                BIT12=  10000
                                BIT11=  4000
                                BIT10=  2000
                                BIT09=  1000
                                BIT08=  400
                                BIT07=  200
                                BIT06=  100
```

```
1016    000040                  BIT05=  40
1017    000020                  BIT04=  20
1018    000010                  BIT03=  10
1019    000004                  BIT02=  4
1020    000002                  BIT01=  2
1021    000001                  BIT00=  1
1022                            .EQUIV  BIT09,BIT9
1023                            .EQUIV  BIT08,BIT8
1024                            .EQUIV  BIT07,BIT7
1025                            .EQUIV  BIT06,BIT6
1026                            .EQUIV  BIT05,BIT5
1027                            .EQUIV  BIT04,BIT4
1028                            .EQUIV  BIT03,BIT3
1029                            .EQUIV  BIT02,BIT2
1030                            .EQUIV  BIT01,BIT1
1031                            .EQUIV  BIT00,BIT0
1032
1033                            ;*BASIC "CPU" TRAP VECTOR ADDRESSES
1034    000004                  ERRVEC= 4               ;;TIME OUT AND OTHER ERRORS
1035    000010                  RESVEC= 10              ;;RESERVED AND ILLEGAL INSTRUCTIONS
1036    000014                  TBITVEC=14              ;;"T" BIT
1037    000014                  TRTVEC= 14              ;;TRACE TRAP
1038    000014                  BPTVEC= 14              ;;BREAKPOINT TRAP (BPT)
1039    000020                  IOTVEC= 20              ;;INPUT/OUTPUT TRAP (IOT) **SCOPE**
1040    000024                  PWRVEC= 24              ;;POWER FAIL
1041    000030                  EMTVEC= 30              ;;EMULATOR TRAP (EMT) **ERROR**
1042    000034                  TRAPVEC=34              ;;"TRAP" TRAP
1043    000060                  TKVEC=  60              ;;TTY KEYBOARD VECTOR
1044    000064                  TPVEC=  64              ;;TTY PRINTER VECTOR
1045    000240                  PIRQVEC=240             ;;PROGRAM INTERRUPT REQUEST VECTOR
1046
1047                            .SBTTL  RM03 REGISTER BIT DEFINITIONS
1048
1049                            ;RMCS1  CONTROL STATUS REGISTER
1050
1051    004000                  DVA     =       BIT11           ;DEVICE AVAILABLE-READ ONLY
1052    000040                  F4      =       BIT05           ;FUNCTION CODE
1053    000020                  F3      =       BIT04           ;FUNCTION CODE
1054    000010                  F2      =       BIT03           ;FUNCTION CODE
1055    000004                  F1      =       BIT02           ;FUNCTION CODE
```

```
1056    000002          FO      =       BIT01           ;FUNCTION CODE
1057    000001          GO      =       BIT00           ;GO BIT
1058    000077          FNCMSK  =       000077          ;FUNCTION CODE MASK
1059
1060                    ;FUNCTION CODES (BITS 01-05 OF RMCS1)
1061
1062    000000          NOP     =       000000          ;NOP COMMAND
1063    000002          ILF02   =       000002          ;ILLEGAL COMMAND
1064    000004          SEEK.   =       000004          ;SEEK COMMAND
1065    000006          RECAL   =       000006          ;RECALIBRATE COMMAND
1066    000010          DRVCLR  =       000010          ;DRIVE CLEAR COMMAND
1067    000012          RLEASE  =       000012          ;RELEASE COMMAND
1068    000014          OFFSET  =       000014          ;OFFSET COMMAND
1069    000016          RTC     =       000016          ;RETURN TO CENTERLINE COMMAND
1070    000020          RIP     =       000020          ;READ IN PRESET COMMAND
1071    000022          PAKACK  =       000022          ;PACK ACKNOWLEDGE COMMAND
1072    000022          PACACK  =       PAKACK
1073    000024          ILF24   =       000024          ;ILLEGAL COMMAND
1074    000026          ILF26   =       000026          ;ILLEGAL COMMAND
1075    000030          SEARCH  =       000030          ;SEARCH COMMAND
1076    000030          ILF30   =       000030          ;ILLEGAL COMMAND
1077    000032          ILF32   =       000032          ;ILLEGAL COMMAND
1078    000034          ILF34   =       000034          ;ILLEGAL COMMAND
1079    000036          ILF36   =       000036          ;ILLEGAL COMMAND
1080    000040          ILF40   =       000040          ;ILLEGAL COMMAND
1081    000042          ILF42   =       000042          ;ILLEGAL COMMAND
1082    000044          ILF44   =       000044          ;ILLEGAL COMMAND
1083    000046          ILF46   =       000046          ;ILLEGAL COMMAND
1084    000050          WCD     =       000050          ;WRITE CHECK DATA COMMAND
1085    000052          WCH     =       000052          ;WRITE CHECK HEADER AND DATA
1086    000054          ILF54   =       000054          ;ILLEGAL COMMAND
1087    000056          ILF56   =       000056          ;ILLEGAL COMMAND
1088    000060          WD      =       000060          ;WRITE DATA COMMAND
1089    000062          WH      =       000062          ;WRITE HEADER AND DATA COMMAND
1090    000064          ILF64   =       000064          ;ILLEGAL COMMAND
1091    000066          ILF66   =       000066          ;ILLEGAL COMMAND
1092    000070          RD      =       000070          ;READ DATA COMMAND
1093    000072          RH      =       000072          ;READ HEADER AND DATA COMMAND
1094    000074          ILF74   =       000074          ;ILLEGAL COMMAND
1095    000076          ILF76   =       000076          ;ILLEGAL COMMAND
1096
1097                    ;RMDA   DISK ADDRESS REGISTER
1098
1099    002000          TA4     =       BIT10           ;TRACK ADDRESS 4
1100    001000          TA2     =       BIT09           ;TRACK ADDRESS 2
1101    000400          TA1     =       BIT08           ;TRACK ADDRESS 1
1102    000020          SA16    =       BIT04           ;SECTOR ADDRESS 16
1103    000010          SA8     =       BIT03           ;SECTOR ADDRESS 8
1104    000004          SA4     =       BIT02           ;SECTOR ADDRESS 4
1105    000002          SA2     =       BIT01           ;SECTOR ADDRESS 2
1106    000001          SA1     =       BIT00           ;SECTOR ADDRESS 1
1107
1108                    ;TRACK,SECTOR MASKS
1109
1110    003400          TADMSK  =       003400          ;TRACK ADDRESS MASK
1111    000037          SADMSK  =       000037          ;SECTOR ADDRESS MASK
```

```
1112
1113                            ;RMDS   DRIVE STATUS REGISTER
1114
1115        100000              ATA     =       BIT15           ;ATTENTION ACTIVE
1116        040000              ERR     =       BIT14           ;COMPOSITE ERROR
1117        020000              PIP     =       BIT13           ;POSITIONING IN PROGRESS
1118        010000              MOL     =       BIT12           ;MEDIUM ON LINE
1119        004000              WRL     =       BIT11           ;WRITE LOCK
1120        002000              LBT     =       BIT10           ;LAST BLOCK TRANSFERRED
1121        001000              PGM     =       BIT09           ;PROGRAMMABLE
1122        000400              DPR     =       BIT08           ;DRIVE PRESENT
1123        000200              DRY     =       BIT07           ;DRIVE READY
1124        000100              VV      =       BIT06           ;VOLUME VALID
1125        000001              OM      =       BIT00           ;OFFSET MODE ACTIVE
1126
1127                            ;RMER1  ERROR REGISTER #1
1128
1129        100000              DCK     =       BIT15           ;DATA CHECK ERROR
1130        040000              UNS     =       BIT14           ;DRIVE UNSAFE
1131        020000              OPI     =       BIT13           ;OPERATION INCOMPLETE
1132        010000              DTE     =       BIT12           ;DRIVE TIMING ERROR
1133        004000              WLE     =       BIT11           ;WRITE LOCK ERROR
1134        002000              IAE     =       BIT10           ;INVALID ADDRESS ERROR
1135        001000              AOE     =       BIT09           ;ADDRESS OVERFLOW ERROR
1136        000400              HCRC    =       BIT08           ;HEADER CRC ERROR
1137        000200              HCE     =       BIT07           ;HEADER COMPARE ERROR
1138        000100              ECH     =       BIT06           ;ECC "HARD" ERROR
1139        000040              WCF     =       BIT05           ;WRITE CLOCK FAILURE
1140        000020              FER     =       BIT04           ;FORMAT ERROR
1141        000010              PAR     =       BIT03           ;PARITY ERROR
1142        000004              RMR     =       BIT02           ;REGISTER MODIFICATION REFUSED
1143        000002              ILR     =       BIT01           ;ILLEGAL REGISTER
1144        000001              ILF     =       BIT00           ;ILLEGAL FUNCTION
1145
1146        115760              NDTMSK  =       DCK!DTE!WLE!AOE!HCRC!HCE!ECH!WCF!FER
1147                            ;"NDTMSK" IS USED TO MASK ERROR REGISTER 1 DURING NON - DATA
1148                            ;COMMANDS, I.E., HOUSEKEEPING AND POSITIONING COMMANDS
1149
1150                            ;RMAS   ATTENTION SUMMARY REGISTER
1151
1152        000377              ATNMSK  =       377             ;MASK FOR ATTENTION BITS
1153
1154                            ;RMLA   LOOK AHEAD REGISTER
1155
1156        002000              SC4     =       BIT10           ;SECTOR COUNT = 16
1157        001000              SC3     =       BIT09           ;SECTOR COUNT = 8
1158        000400              SC2     =       BIT08           ;SECTOR COUNT = 4
1159        000200              SC1     =       BIT07           ;SECTOR COUNT = 2
1160        000100              SC0     =       BIT06           ;SECTOR COUNT = 1
1161
1162        003700              SCTMSK  =       003700          ;SECTOR COUNT MASK
1163
1164                            ;RMMR   MAINTENANCE REGISTER
1165
1166                            ;       WRITE ONLY BITS
1167
```

```
1168          100000                  DBCK    =       BIT15           ;DEBUG CLOCK
1169          040000                  DBEN    =       BIT14           ;DEBUG CLOCK ENABLE
```

```
1170            020000          DEBL    =       BIT13           ;DIAGNOSTIC END OF BLOCK
1171            010000          DTO     =       BIT12           ;DIAGNOSTIC TIMEOUT
1172            004000          MCLK    =       BIT11           ;MAINTENANCE CLOCK
1173            002000          MRD     =       BIT10           ;READ DATA
1174            001000          MUR     =       BIT09           ;UNIT READY
1175            000400          MOC     =       BIT08           ;ON CYLINDER
1176            000200          MSER    =       BIT07           ;SEEK ERROR
1177            000100          MDF     =       BIT06           ;DRIVE FAULT
1178            000040          MS      =       BIT05           ;SECTOR PULSE
1179            000010          MWP     =       BIT03           ;WRITE PROTECT
1180            000004          MI      =       BIT02           ;INDEX PULSE
1181            000002          MSC     =       BIT01           ;SECTOR COMPARE
1182            000001          DMD     =       BIT00           ;DIAGNOSTIC MODE
1183            051401          MR1AAA  =       DMD!MUR!DBEN!MOC!DTO
1184
1185                            ;       READ ONLY BITS
1186
1187            100000          OCC     =       BIT15           ;OCCUPIED
1188            040000          RG      =       BIT14           ;RUN AND GO
1189            020000          EBL     =       BIT13           ;END OF BLOCK
1190            010000          REX     =       BIT12           ;EXCEPTION
1191            004000          ESRC    =       BIT11           ;ENABLE SEARCH
1192            002000          PLFS    =       BIT10           ;LOOKING FOR SYNC
1193            001000          ECRC    =       BIT09           ;ENABLE CRC OUT
1194            000400          PDA     =       BIT08           ;DATA AREA
1195            000200          PHA     =       BIT07           ;HEADER AREA
1196            000100          CONT    =       BIT06           ;CONTINUE
1197            000040          WC      =       BIT05           ;WORD CLOCK
1198            000020          EECC    =       BIT04           ;ENABLE ECC OUT
1199            000010          MWD     =       BIT03           ;WRITE DATA BIT
1200            000004          LS      =       BIT02           ;LAST SECTOR
1201            000002          LST     =       BIT01           ;LAST SECTOR AND TRACK
1202            000001          DMD     =       BIT00           ;DIAGNOSTIC MODE
1203
1204                            ;RMDT   DRIVE TYPE REGISTER
1205
1206            100000          NSA     =       BIT15           ;NOT SECTOR ADDRESSED=0
1207            040000          TAP     =       BIT14           ;TAPE DRIVE = 0
1208            020000          MOH     =       BIT13           ;MOVING HEAD = 1
1209            004000          DRQ     =       BIT11           ;DRIVE REQUEST REQUIRED
1210
1211            020024          SNGPRT  =       020024          ;SINGLE PORT DRIVE TYPE
1212            024024          DULPRT  =       024024          ;DUAL PORT DRIVE TYPE
1213
1214                            ;RMOF   OFFSET REGISTER
1215
1216            010000          FMT16   =       BIT12           ;16 BIT WORD FORMAT
1217            004000          ECI     =       BIT11           ;ECC INHIBIT
1218            002000          HCI     =       BIT10           ;HEADER COMPARE INHIBIT
1219            000200          OFD     =       BIT07           ;OFFSET FORWARD
1220            161577          XNUOF   =       161577          ;UNUSED BITS OF RMOF
1221
1222
1223                            ;RMDC   DESIRED CYLINDER ADDRESS REGISTER
1224            001777          CYLMSK  =       1777            ;MASK FOR CYLINDER ADDRESS
1225            176000          XNUDC   =       176000          ;UNUSED BITS OF RMDC
```

```
1226
1227                                    ;RMMR2  MAINTENANCE REGISTER #2
1228
1229                                    ;       READ ONLY BITS
1230            100000                  RQA     =       BIT15           ;PORT A REQUEST
1231            040000                  RQB     =       BIT14           ;PORT B REQUEST
1232            020000                  TAG     =       BIT13           ;TAG CONTROL
1233            010000                  TST     =       BIT12           ;COMMAND SEQUENCE TEST BIT
1234            004000                  CC      =       BIT11           ;CONTROL OR CYLINDER TAG
1235            002000                  CH      =       BIT10           ;CONTROL OR HEAD TAG
1236            001000                  BB09    =       BIT09           ;TAG BUS
1237            000400                  BB08    =       BIT08           ;TAG BUS
1238            000200                  BB07    =       BIT07           ;TAG BUS
1239            000100                  BB06    =       BIT06           ;TAG BUS
1240            000040                  BB05    =       BIT05           ;TAG BUS
1241            000020                  BB04    =       BIT04           ;TAG BUS
1242            000010                  BB03    =       BIT03           ;TAG BUS
1243            000004                  BB02    =       BIT02           ;TAG BUS
1244            000002                  BB01    =       BIT01           ;TAG BUS
1245            000001                  BB00    =       BIT00           ;TAG BUS
1246
1247
1248                                    ;RMER2  ERROR REGISTER 2
1249
1250            100000                  BSE     =       BIT15           ;BAD SECTOR ERROR
1251            040000                  SKI     =       BIT14           ;SEEK INCOMPLETE
1252            020000                  OPE     =       BIT13           ;OPERATOR PLUG ERROR
1253            010000                  IVC     =       BIT12           ;INVALID COMMAND ERROR
1254            004000                  LSC     =       BIT11           ;LOSS OF SYSTEM CLOCK
1255            002000                  LBC     =       BIT10           ;LOSS OF BIT CLOCK
1256            000200                  DVC     =       BIT07           ;DEVICE CHECK
1257            000010                  DPE     =       BIT03           ;DATA PARITY ERROR
1258            001567                  XNUER2  =       001567          ;UNUSED BITS OF RMER2
1259
1260                                    .SBTTL  PROGRAM MNEMONICS
1261
1262            100000                  MSE     =       BIT15           ;MANUFACTURING DETECTED SECTOR ERROR
1263            040000                  USE     =       BIT14           ;USER DETECTED SECTOR ERROR
1264
1265                                    .SBTTL  RM03 REGISTER INDEX VALUES
1266
1267            000000                  RMCS1   =       00              ;CONTROL STATUS REGISTER
1268            000006                  RMDA    =       06              ;DISK ADDRESS REGISTER
1269            000012                  RMDS    =       12              ;DRIVE STATUS REGISTER
1270            000014                  RMER1   =       14              ;ERROR REGISTER 1
1271            000016                  RMAS    =       16              ;ATTENTION SUMMARY REGISTER
1272            000020                  RMLA    =       20              ;LOOK AHEAD REGISTER
1273            000024                  RMMR1   =       24              ;MAINTENANCE REGISTER
1274            000026                  RMDT    =       26              ;DRIVE TYPE REGISTER
1275            000030                  RMSN    =       30              ;SERIAL NUMBER REGISTER
1276            000032                  RMOF    =       32              ;OFFSET REGISTER
1277            000034                  RMDC    =       34              ;DESIRED CYLINDER REGISTER
1278            000036                  RMHR    =       36              ;HOLDING REGISTER
1279            000040                  RMMR2   =       40              ;MAINTENANCE REGISTER 2
1280            000042                  RMER2   =       42              ;ERROR REGISTER 2
1281            000044                  RMEC1   =       44              ;ECC POSITION REGISTER
```

```
1282    000046          RMEC2   =       46              ;ECC PATTERN REGISTER
1283    000050          ILRG50  =       50              ;ILLEGAL REGISTER 50
1284    000052          ILRG52  =       52              ;ILLEGAL REGISTER 52
1285    000054          ILRG54  =       54              ;ILLEGAL REGISTER 54
1286    000056          ILRG56  =       56              ;ILLEGAL REGISTER 56
1287    000060          ILRG60  =       60              ;ILLEGAL REGISTER 60
1288    000062          ILRG62  =       62              ;ILLEGAL REGISTER 62
1289    000064          ILRG64  =       64              ;ILLEGAL REGISTER 64
1290    000066          ILRG66  =       66              ;ILLEGAL REGISTER 66
1291    000070          ILRG70  =       70              ;ILLEGAL REGISTER 70
1292    000072          ILRG72  =       72              ;ILLEGAL REGISTER 72
1293    000074          ILRG74  =       74              ;ILLEGAL REGISTER 74
1294    000076          ILRG76  =       76              ;ILLEGAL REGISTER 76
1295
1296
1297    000077          IDXMSK  =       77              ;MASK FOR REGISTER INDEX NUMBER
1298
1299                    .SBTTL  RH CONTROLLER REGISTER BIT DEFINITIONS
1300
1301                    ;RMCS1  CONTROL STATUS REGISTER #1
1302
1303    100000          SC      =       BIT15           ;SPECIAL CONDITION-READ ONLY
1304    040000          TRE     =       BIT14           ;TRANSFER ERROR
1305    020000          MCPE    =       BIT13           ;MASSBUS CONTROL BUS PARITY
1306                                                    ;ERROR-READ ONLY
1307    002000          PSEL    =       BIT10           ;PORT B SELECT
1308    001000          A17     =       BIT09           ;ADDRESS EXTENSION
1309    000400          A16     =       BIT08           ;ADDRESS EXTENSION
1310    000200          RDY     =       BIT07           ;READY-READ ONLY
1311    000100          IE      =       BIT06           ;INTERRUPT ENABLE
1312
1313                    ;RMCS2  RH CONTROL STATUS REGISTER #2
1314
1315    100000          DLT     =       BIT15           ;DATA LATE-READ ONLY
1316    040000          WCE     =       BIT14           ;WRITE CHECK ERROR-READ ONLY
1317    020000          UPE     =       BIT13           ;UNIBUS PARITY ERROR
1318    010000          NED     =       BIT12           ;NONEXISTANT DRIVE-READ ONLY
1319    004000          NEM     =       BIT11           ;NONEXISTANT MEMORY-READ ONLY
1320    002000          PGE     =       BIT10           ;PROGRAM ERROR-READ ONLY
1321    001000          MXF     =       BIT09           ;MISSED TRANSFER
1322    000400          MDPE    =       BIT08           ;MASSBUS DATA BUS PARITY
1323                                                    ;ERROR-READ ONLY
1324    000200          OR      =       BIT07           ;OUTPUT READY-READ ONLY
1325    000100          IR      =       BIT06           ;INPUT READY-READ ONLY
1326    000040          CLR     =       BIT05           ;CONTROLLER CLEAR
1327    000020          PAT     =       BIT04           ;PARITY TEST
1328    000010          BAI     =       BIT03           ;UNIBUS ADDRESS INCREMENT
1329                                                    ;INHIBIT
1330    000004          U2      =       BIT02           ;UNIT SELECT
1331    000002          U1      =       BIT01           ;UNIT SELECT
1332    000001          U0      =       BIT00           ;UNIT SELECT
1333
1334                    ;UNIT SELECT MASK
1335    000007          UNTMSK  =       7               ;UNIT SELECT MASK
1336
1337                    ;RMCS3  RH70 CONTROL STATUS REGISTER #3
```

```
1338            100000          APE     =       BIT15           ;ADDRESS PARITY ERROR
1339            040000          DPEHI   =       BIT14           ;DATA PARITY ERROR HIGH WORD
1340            020000          DPELO   =       BIT13           ;DATA PARITY ERROR LOW WORD
1341            010000          WCEHI   =       BIT12           ;WRITE CHECK ERROR HIGH WORD
1342            004000          WCELO   =       BIT11           ;WRITE CHECK ERROR LOW WORD
1343            002000          DBL     =       BIT10           ;DOUBLE WORD TRANSFER
1344            000100          IE      =       BIT06           ;INTERRUPT ENABLE
1345            000010          IPCK3   =       BIT03           ;INVERT PARITY CHECK
1346            000004          IPCK2   =       BIT02           ;INVERT PARITY CHECK
1347            000002          IPCK1   =       BIT01           ;INVERT PARITY CHECK
1348            000001          IPCK0   =       BIT00           ;INVERT PARITY CHECK
1349                            .SBTTL  RH CONTROLLER REGISTER INDEX VALUES
1350
1351            000000          RMCS1   =       00              ;CONTROL, STATUS REGISTER
1352            000002          RMWC    =       02              ;WORD COUNT REGISTER
1353            000004          RMBA    =       04              ;BUS ADDRESS REGISTER
1354            000010          RMCS2   =       10              ;CONTROLLER STATUS REGISTER
1355            000022          RMDB    =       22              ;DATA BUFFER
1356            000050          RMBAE   =       50              ;BUS ADDRESS EXTENSION
1357            000052          RMCS3   =       52              ;CONTROL STATUS REGISTER #3
1358
1359            176700          ABASE   =       176700          ;UNIBUS ADDRESS
1360            120254          AVECT1  =       120254          ;UNIBUS VECTOR ADDRESS AND PRIORITY
1361
1362
1363                            .SBTTL  TRAP CATCHER
1364
1365            000000                  .=0
1366                            ;*ALL UNUSED LOCATIONS FROM 4 - 776 CONTAIN A ".+2,HALT"
1367                            ;*SEQUENCE TO CATCH ILLEGAL TRAPS AND INTERRUPTS
1368                            ;*LOCATION 0 CONTAINS 0 TO CATCH IMPROPERLY LOADED VECTORS
1369            000174                  .=174
1370    000174  000000          DISPREG: .WORD  0                ;;SOFTWARE DISPLAY REGISTER
1371    000176  000000          SWREG:   .WORD  0                ;;SOFTWARE SWITCH REGISTER
1372
1373
1374                            .SBTTL  ACT11 HOOKS
1375
1376                            ;;******************************************************************
1377                            ;HOOKS REQUIRED BY ACT11
1378            000200                  $SVPC=.                  ;SAVE PC
1379            000046                  .=46
1380    000046  021642                  $ENDAD                   ;;1)SET LOC.46 TO ADDRESS OF $ENDAD IN .$EOP
1381            000052                  .=52
1382    000052  000000                  .WORD   0                ;;2)SET LOC.52 TO ZERO
1383            000200                  .=$SVPC                  ;; RESTORE PC
1384
1385                            .SBTTL  STARTING ADDRESS
1386
1387                            ;THE PROGRAM STARTS AT LOCATION 200
1388            000200          .       =       200
1389    000200  000137  002632          JMP     START           ;JUMP TO START OF PROGRAM
1390
1391            001100          .=1100
1392                            .SBTTL  APT PARAMETER BLOCK
1393
```

```
1394                                    ;;*****************************************************************
1395                                    ;SET LOCATIONS 24 AND 44 AS REQUIRED FOR APT
1396                                    ;;*****************************************************************
1397            001100                          .$X=.   ;;SAVE CURRENT LOCATION
1398            000024                          .=24    ;;SET POWER FAIL TO POINT TO START OF PROGRAM
1399   000024   000200                          200     ;;FOR APT START UP
1400            000044                          .=44    ;;POINT TO APT INDIRECT ADDRESS PNTR.
1401   000044   001100                          $APTHDR ;;POINT TO APT HEADER BLOCK
1402            001100                          .=.$X   ;;RESET LOCATION COUNTER
1403                                    ;;*****************************************************************
1404                                    ;SETUP APT PARAMETER BLOCK AS DEFINED IN THE APT-PDP11 DIAGNOSTIC
1405                                    ;INTERFACE SPEC.
1406
1407   001100                           $APTHD:
1408   001100   000000                 $HIBTS: .WORD   0       ;;TWO HIGH BITS OF 18 BIT MAILBOX ADDR.
1409   001102   001222                 $MBADR: .WORD   $MAIL   ;;ADDRESS OF APT MAILBOX (BITS 0-15)
1410   001104   000001                 $TSTM:  .WORD   1       ;;RUN TIM OF LONGEST TEST
1411   001106   000002                 $PASTM: .WORD   2       ;;RUN TIME IN SECS. OF 1ST PASS ON 1 UNIT (QUICK VERIFY)
1412   001110   000002                 $UNITM: .WORD   2       ;;ADDITIONAL RUN TIME (SECS) OF A PASS FOR EACH ADDITIONAL UNIT
1413   001112   000042                         .WORD   $ETEND-$MAIL/2 ;;LENGTH MAILBOX-ETABLE(WORDS)
1414            001114                 TAGADR  =       .
```

```
                                .SBTTL  COMMON TAGS

                                ;;*****************************************************************
                                ;*THIS TABLE CONTAINS VARIOUS COMMON STORAGE LOCATIONS
                                ;*USED IN THE PROGRAM.

                        001114          .=TAGADR
001114                  $CMTAG:                         ;;START OF COMMON TAGS
001114  000000                  .WORD   0
001116     000          $TSTNM: .BYTE   0               ;;CONTAINS THE TEST NUMBER
001117     000          $ERFLG: .BYTE   0               ;;CONTAINS ERROR FLAG
001120  000000          $ICNT:  .WORD   0               ;;CONTAINS SUBTEST ITERATION COUNT
001122  000000          $LPADR: .WORD   0               ;;CONTAINS SCOPE LOOP ADDRESS
001124  000000          $LPERR: .WORD   0               ;;CONTAINS SCOPE RETURN FOR ERRORS
001126  000000          $ERTTL: .WORD   0               ;;CONTAINS TOTAL ERRORS DETECTED
001130     000          $ITEMB: .BYTE   0               ;;CONTAINS ITEM CONTROL BYTE
001131     001          $ERMAX: .BYTE   1               ;;CONTAINS MAX. ERRORS PER TEST
001132  000000          $ERRPC: .WORD   0               ;;CONTAINS PC OF LAST ERROR INSTRUCTION
001134  000000          $GDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'GOOD' DATA
001136  000000          $BDADR: .WORD   0               ;;CONTAINS ADDRESS OF 'BAD' DATA
001140  000000          $GDDAT: .WORD   0               ;;CONTAINS 'GOOD' DATA
001142  000000          $BDDAT: .WORD   0               ;;CONTAINS 'BAD' DATA
001144  000000                  .WORD   0               ;;RESERVED--NOT TO BE USED
001146  000000                  .WORD   0
001150     000          $AUTOB: .BYTE   0               ;;AUTOMATIC MODE INDICATOR
001151     000          $INTAG: .BYTE   0               ;;INTERRUPT MODE INDICATOR
001152  000000                  .WORD   0
001154  177570          SWR:    .WORD   DSWR            ;;ADDRESS OF SWITCH REGISTER
001156  177570          DISPLAY: .WORD  DDISP           ;;ADDRESS OF DISPLAY REGISTER
001160  177560          $TKS:   177560                  ;;TTY KBD STATUS
001162  177562          $TKB:   177562                  ;;TTY KBD BUFFER
001164  177564          $TPS:   177564                  ;;TTY PRINTER STATUS REG. ADDRESS
001166  177566          $TPB:   177566                  ;;TTY PRINTER BUFFER REG. ADDRESS
001170     000          $NULL:  .BYTE   0               ;;CONTAINS NULL CHARACTER FOR FILLS
001171     002          $FILLS: .BYTE   2               ;;CONTAINS # OF FILLER CHARACTERS REQUIRED
001172     012          $FILLC: .BYTE   12              ;;INSERT FILL CHARS. AFTER A "LINE FEED"
001173     000          $TPFLG: .BYTE   0               ;;"TERMINAL AVAILABLE" FLAG (BIT<07>=0=YES)
001174  000000          $TMP0:  .WORD   0               ;;USER DEFINED
001176  000000          $TMP1:  .WORD   0               ;;USER DEFINED
001200  000000          $TMP2:  .WORD   0               ;;USER DEFINED
001202  000000          $TMP3:  .WORD   0               ;;USER DEFINED
001204  000000          $TMP4:  .WORD   0               ;;USER DEFINED
001206  000000          $TIMES: 0                       ;;MAX. NUMBER OF ITERATIONS
001210  000000          $ESCAPE:0                       ;;ESCAPE ON ERROR ADDRESS
001212  177607  000377    $BELL:  .ASCIZ  <207><377><377> ;;CODE FOR BELL
001216     077          $QUES:  .ASCII  /?/             ;;QUESTION MARK
001217     015          $CRLF:  .ASCII  <15>            ;;CARRIAGE RETURN
001220  000012          $LF:    .ASCIZ  <12>            ;;LINE FEED
                                ;;*****************************************************************
                                .SBTTL  APT MAILBOX-ETABLE

                                ;;*****************************************************************
                                .NLIST ME
                                ;
                                ;
```

```
 1471                                   .EVEN
 1472  001222                           $MAIL:                  ;;APT MAILBOX
 1473  001222  000000                   $MSGTY: .WORD   AMSGTY  ;;MESSAGE TYPE CODE
 1474  001224  000000                   $FATAL: .WORD   AFATAL  ;;FATAL ERROR NUMBER
 1475  001226  000000                   $TESTN: .WORD   ATESTN  ;;TEST NUMBER
 1476  001230  000000                   $PASS:  .WORD   APASS   ;;PASS COUNT
 1477  001232  000000                   $DEVCT: .WORD   ADEVCT  ;;DEVICE COUNT
 1478  001234  000000                   $UNIT:  .WORD   AUNIT   ;;I/O UNIT NUMBER
 1479  001236  000000                   $MSGAD: .WORD   AMSGAD  ;;MESSAGE ADDRESS
 1480  001240  000000                   $MSGLG: .WORD   AMSGLG  ;;MESSAGE LENGTH
 1481  001242                           $ETABLE:                ;;APT ENVIRONMENT TABLE
 1482  001242     000                   $ENV:   .BYTE   AENV    ;;ENVIRONMENT BYTE
 1483  001243     000                   $ENVM:  .BYTE   AENVM   ;;ENVIRONMENT MODE BITS
 1484  001244  000000                   $SWREG: .WORD   ASWREG  ;;APT SWITCH REGISTER
 1485  001246  000000                   $USWR:  .WORD   AUSWR   ;;USER SWITCHES
 1486  001250  000000                   $CPUOP: .WORD   ACPUOP  ;;CPU TYPE,OPTIONS
 1487                                   ;*                      BITS 15-11=CPU TYPE
 1488                                   ;*                              11/04=01,11/05=02,11/20=03,11/40=04,11/45=05
 1489                                   ;*                              11/70=06,PDQ=07,Q=10
 1490                                   ;*                      BIT 10=REAL TIME CLOCK
 1491                                   ;*                      BIT  9=FLOATING POINT PROCESSOR
 1492                                   ;*                      BIT  8=MEMORY MANAGEMENT
 1493  001252     000                   $MAMS1: .BYTE   AMAMS1  ;;HIGH ADDRESS,M.S. BYTE
 1494  001253     000                   $MTYP1: .BYTE   AMTYP1  ;;MEM. TYPE,BLK#1
 1495                                   ;*                      MEM.TYPE BYTE   --  (HIGH BYTE)
 1496                                   ;*                              900 NSEC CORE=001
 1497                                   ;*                              300 NSEC BIPOLAR=002
 1498                                   ;*                              500 NSEC MOS=003
 1499  001254  000000                   $MADR1: .WORD   AMADR1  ;;HIGH ADDRESS,BLK#1
 1500                                   ;*                        MEM.LAST ADDR.=3 BYTES,THIS WORD AND LOW OF ''TYPE'' ABOVE
 1501  001256     000                   $MAMS2: .BYTE   AMAMS2  ;;HIGH ADDRESS,M.S. BYTE
 1502  001257     000                   $MTYP2: .BYTE   AMTYP2  ;;MEM.TYPE,BLK#2
 1503  001260  000000                   $MADR2: .WORD   AMADR2  ;;MEM.LAST ADDRESS,BLK#2
 1504  001262     000                   $MAMS3: .BYTE   AMAMS3  ;;HIGH ADDRESS,M.S.BYTE
 1505  001263     000                   $MTYP3: .BYTE   AMTYP3  ;;MEM.TYPE,BLK#3
 1506  001264  000000                   $MADR3: .WORD   AMADR3  ;;MEM.LAST ADDRESS,BLK#3
 1507  001266     000                   $MAMS4: .BYTE   AMAMS4  ;;HIGH ADDRESS,M.S.BYTE
 1508  001267     000                   $MTYP4: .BYTE   AMTYP4  ;;MEM.TYPE,BLK#4
 1509  001270  000000                   $MADR4: .WORD   AMADR4  ;;MEM.LAST ADDRESS,BLK#4
 1510  001272  120254                   $VECT1: .WORD   AVECT1  ;;INTERRUPT VECTOR#1,BUS PRIORITY#1
 1511  001274  000000                   $VECT2: .WORD   AVECT2  ;;INTERRUPT VECTOR#2BUS PRIORITY#2
 1512  001276  176700                   $BASE:  .WORD   ABASE   ;;BASE ADDRESS OF EQUIPMENT UNDER TEST
 1513  001300  000000                   $DEVM:  .WORD   ADEVM   ;;DEVICE MAP
 1514  001302  000000                   $CDW1:  .WORD   ACDW1   ;;CONTROLLER DESCRIPTION WORD#1
 1515  001304  000000                   $CDW2:  .WORD   ACDW2   ;;CONTROLLER DESCRIPTION WORD#2
 1516  001306  000000                   $DDW0:  .WORD   ADDW0   ;;DEVICE DESCRIPTOR WORD#0
 1517  001310  000000                   $DDW1:  .WORD   ADDW1   ;;DEVICE DESCRIPTOR WORD#1
 1518  001312  000000                   $DDW2:  .WORD   ADDW2   ;;DEVICE DESCRIPTOR WORD#2
 1519  001314  000000                   $DDW3:  .WORD   ADDW3   ;;DEVICE DESCRIPTOR WORD#3
 1520  001316  000000                   $DDW4:  .WORD   ADDW4   ;;DEVICE DESCRIPTOR WORD#4
 1521  001320  000000                   $DDW5:  .WORD   ADDW5   ;;DEVICE DESCRIPTOR WORD#5
 1522  001322  000000                   $DDW6:  .WORD   ADDW6   ;;DEVICE DESCRIPTOR WORD#6
 1523  001324  000000                   $DDW7:  .WORD   ADDW7   ;;DEVICE DESCRIPTOR WORD#7
 1524  001326                           $ETEND:
 1525                                   .MEXIT
 1526
```

```
1527
1528
1529                                    .SBTTL  REGISTER INPUT BUFFER
1530
1531                                    ;       THE REGISTER INPUT BUFFER IS USED FOR STORING REGISTER
1532                                    ;CONTENTS AS THEY ARE READ FROM THE DEVICE.
1533
1534  001326  000000                    RMCS1I: .WORD                   ;CONTROL,STATUS REGISTER #1
1535  001330  000000                    RMWCI: .WORD                    ;WORD COUNT REGISTER
1536  001332  000000                    RMBAI: .WORD                    ;BUS ADDRESS REGISTER
1537  001334  000000                    RMDAI: .WORD                    ;DISK ADDRESS REGISTER
1538  001336  000000                    RMCS2I: .WORD                   ;CONTROL, STATUS REGISTER #2
1539  001340  000000                    RMDSI: .WORD                    ;DRIVE STATUS REGISTER
1540  001342  000000                    RMER1I: .WORD                   ;ERROR REGISTER 1
1541  001344  000000                    RMASI: .WORD                    ;ATTENTION SUMMARY REGISTER
1542  001346  000000                    RMLAI: .WORD                    ;LOOK AHEAD REGISTER
1543  001350  000000                    RMDBI: .WORD                    ;DATA BUFFER
1544  001352  000000                    RMMR1I: .WORD                   ;MAINTENANCE REGISTER #1
1545  001354  000000                    RMDTI: .WORD                    ;DRIVE TYPE REGISTER
1546  001356  000000                    RMSNI: .WORD                    ;SERIAL NUMBER REGISTER
1547  001360  000000                    RMOFI: .WORD                    ;OFFSET REGISTER
1548  001362  000000                    RMDCI: .WORD                    ;DESIRED CYLINDER REGISTER
1549  001364  000000                    RMHRI: .WORD                    ;HOLDING REGISTER
1550  001366  000000                    RMMR2I: .WORD                   ;MAINTENANCE REGISTER #2
1551  001370  000000                    RMER2I: .WORD                   ;ERROR REGISTER 2
1552  001372  000000                    RMEC1I: .WORD                   ;ECC POSITION REGISTER
1553  001374  000000                    RMEC2I: .WORD                   ;ECC PATTERN REGISTER
1554  001376  000000                    RMBAEI: .WORD           ;BUS ADDRESS EXTENSION REGISTER
1555  001400  000000                    RMCS3I: .WORD           ;CONTROL, STATUS REGISTER #3
1556
1557
1558
1559                                    .SBTTL  REGISTER OUTPUT BUFFER
1560
1561                                    ;       THE REGISTER OUTPUT BUFFER IS USED FOR ASSEMBLING DATA TO
1562                                    ;BE WRITTEN TO THE DEVICE.
1563
1564  001402  000000                    RMCS1O: .WORD                   ;CONTROL,STATUS REGISTER #1
1565  001404  000000                    RMWCO: .WORD                    ;WORD COUNT REGISTER
1566  001406  000000                    RMBAO: .WORD                    ;BUS ADDRESS REGISTER
1567  001410  000000                    RMDAO: .WORD                    ;DISK ADDRESS REGISTER
1568  001412  000000                    RMCS2O: .WORD                   ;CONTROL, STATUS REGISTER #2
1569  001414  000000                    RMDSO: .WORD                    ;DRIVE STATUS REGISTER
1570  001416  000000                    RMER1O: .WORD                   ;ERROR REGISTER 1
1571  001420  000000                    RMASO: .WORD                    ;ATTENTION SUMMARY REGISTER
1572  001422  000000                    RMLAO: .WORD                    ;LOOK AHEAD REGISTER
1573  001424  000000                    RMDBO: .WORD                    ;DATA BUFFER
1574  001426  000000                    RMMR1O: .WORD                   ;MAINTENANCE REGISTER #1
1575  001430  000000                    RMDTO: .WORD                    ;DRIVE TYPE REGISTER
1576  001432  000000                    RMSNO: .WORD                    ;SERIAL NUMBER REGISTER
1577  001434  000000                    RMOFO: .WORD                    ;OFFSET REGISTER
1578  001436  000000                    RMDCO: .WORD                    ;DESIRED CYLINDER REGISTER
1579  001440  000000                    RMHRO: .WORD                    ;HOLDING REGISTER
1580  001442  000000                    RMMR2O: .WORD                   ;MAINTENANCE REGISTER #2
1581  001444  000000                    RMER2O: .WORD                   ;ERROR REGISTER 2
1582  001446  000000                    RMEC1O: .WORD                   ;ECC POSITION REGISTER
```

```
1583  001450  000000                  RMEC20: .WORD                   ;ECC PATTERN REGISTER
1584  001452  000000                  RMBAE0: .WORD           ;BUS ADDRESS EXTENSION REGISTER
1585  001454  000000                  RMCS30: .WORD           ;CONTROL, STATUS REGISTER #3
1586
1587                                  .SBTTL  TEST QUE
1588
1589                                  ;       EACH WORD OF THE TEST QUE CONTAINS THE DEVICE NUMBER IN
1590                                  ;THE LOW BYTE AND THE ATTENTION BIT IN THE HIGH BYTE. A ZERO
1591                                  ;WORD IS A BLANK AND REPRESENTS THE END OF THE QUE.
1592
1593  001456  000012                  TSTQUE: .BLKW   10.     ;TEST QUE
1594
1595  001502  172540                  $LPCSR: .WORD   172540          ;KW11-P CONTROL + STATUS REGISTER
1596  001504  172542                  $LPCSB: .WORD   172542          ;KW11-P COUNT SET BUFFER
1597  001506  000104                  $LPVEC: .WORD   104             ;KW11-P INTERRUPT VECTOR
1598  001510  000106                          .WORD   106
1599  001512  177546                  $LLCSR: .WORD   177546          ;KW11-L CONTROL + STATUS REGISTER
1600  001514  000100                  $LLVEC: .WORD   100             ;KW11-L INTERRUPT VECTOR
1601  001516  000102                          .WORD   102
1602  001520  000000                  $PSW:   .WORD                   ;STORAGE FOR PRIORITY
1603  001522  000000                  TIME:   .WORD                   ;STORAGE FOR ELAPSED TIME
1604  001524  000000                  WATCH:  .WORD                   ;STORAGE FOR REMAINING TIME
1605  001526  000000                  CLOCK:  .WORD                   ;ADDRESS OF START CLOCK SUB
1606  001530  000000                  STOP:   .WORD                   ;ADDRESS OF STOP CLOCK SUB
1607
1608                                  ;PUT TAGS HERE
1609
```

```
                                .SBTTL  ERROR POINTER TABLE

                                ;*THIS TABLE CONTAINS THE INFORMATION FOR EACH ERROR THAT CAN OCCUR.
                                ;*THE INFORMATION IS OBTAINED BY USING THE INDEX NUMBER FOUND IN
                                ;*LOCATION $ITEMB. THIS NUMBER INDICATES WHICH ITEM IN THE TABLE IS PERTINENT.
                                ;*NOTE1:        IF $ITEMB IS 0 THE ONLY PERTINENT DATA IS ($ERRPC).
                                ;*NOTE2:        EACH ITEM IN THE TABLE CONTAINS 4 POINTERS EXPLAINED AS FOLLOWS:

                                ;*      EM              ;;POINTS TO THE ERROR MESSAGE
                                ;*      DH              ;;POINTS TO THE DATA HEADER
                                ;*      DT              ;;POINTS TO THE DATA
                                ;*      DF              ;;POINTS TO THE DATA FORMAT


001532                          $ERRTB:
```

```
1627
1628                                    ;ERROR  1       CANNOT CLEAR NED STATUS
1629
1630  001532  031710                            EMT1
1631  001534  037614                            EHT1
1632  001536  037724                            EDT1
1633  001540  037760                            EFT1
1634
1635
1636                                    ;ERROR  2       CANNOT READ OR WRITE ANY DEVICE REG WITHOUT NED
1637
1638  001542  031716                            EMT2
1639  001544  037620                            EHT2
1640  001546  037726                            EDT2
1641  001550  037762                            EFT2
1642
1643
1644                                    ;ERROR  3       CANNOT WRITE/READ ONES TO ANY DEVICE REGISTER
1645
1646  001552  031744                            EMT3
1647  001554  000000                            0
1648  001556  000000                            0
1649  001560  000000                            0
1650
1651
1652                                    ;ERROR  4       CANNOT CLEAR ANY DEVICE REGISTER BITS W/MASSBUS INIT
1653
1654  001562  031764                            EMT4
1655  001564  000000                            0
1656  001566  000000                            0
1657  001570  000000                            0
1658
1659
1660                                    ;ERROR  5       CANNOT WRITE/READ ZEROS TO ALL BIT POSITIONS
1661
1662  001572  032006                            EMT5
1663  001574  037624                            EHT5
1664  001576  037730                            EDT5
1665  001600  037764                            EFT5
1666
1667
1668                                    ;ERROR  6       CANNOT WRITE/READ ONES TO ALL BIT POSITIONS
1669
1670  001602  032032                            EMT6
1671  001604  037624                            EHT5
1672  001606  037730                            EDT5
1673  001610  037764                            EFT5
1674
1675
1676                                    ;ERROR  7       CANNOT WRITE/READ SHIFTING ONE BIT TO ALL BIT POSITIONS
1677                                    ;               OF DEVICE REGISTERS
1678
1679  001612  032054                            EMT7
1680  001614  037630                            EHT7
1681  001616  037730                            EDT5
1682  001620  037764                            EFT5
```

```
1683
1684
1685                                    ;ERROR  10      REGISTER SELECT 1 APPEARS S-A-0
1686
1687  001622  032076                              EMT10
1688  001624  000000                              0
1689  001626  000000                              0
1690  001630  000000                              0
1691
1692
1693                                    ;ERROR  11      REGISTER SELECT 1 APPEARS S-A-1
1694
1695  001632  032114                              EMT11
1696  001634  000000                              0
1697  001636  000000                              0
1698  001640  000000                              0
1699
1700
1701                                    ;ERROR  12      REGISTER SELECT 2 APPEARS S-A-0
1702
1703  001642  032132                              EMT12
1704  001644  000000                              0
1705  001646  000000                              0
1706  001650  000000                              0
1707
1708
1709                                    ;ERROR  13      REGISTER SELECT 2 APPEARS S-A-1
1710
1711  001652  032150                              EMT13
1712  001654  000000                              0
1713  001656  000000                              0
1714  001660  000000                              0
1715
1716
1717                                    ;ERROR  14      REGISTER SELECT 4 APPEARS S-A-0
1718
1719  001662  032166                              EMT14
1720  001664  000000                              0
1721  001666  000000                              0
1722  001670  000000                              0
1723
1724
1725                                    ;ERROR  15      REGISTER SELECT 4 APPEARS S-A-1
1726
1727  001672  032204                              EMT15
1728  001674  000000                              0
1729  001676  000000                              0
1730  001700  000000                              0
1731
1732
1733                                    ;ERROR  16      REGISTER SELECT 8 APPEARS S-A-0
1734
1735  001702  032222                              EMT16
1736  001704  000000                              0
1737  001706  000000                              0
1738  001710  000000                              0
```

```
1739
1740
1741                                    ;ERROR  17      REGISTER SELECT 8 APPEARS S-A-1
1742
1743  001712  032240                            EMT17
1744  001714  000000                            0
1745  001716  000000                            0
1746  001720  000000                            0
1747
1748
1749                                    ;ERROR  20      OPI SET WITH SEACH TIMOUT DISABLED
1750
1751  001722  032256                            EMT20
1752  001724  037614                            EHT1
1753  001726  037724                            EDT1
1754  001730  037760                            EFT1
1755
1756
1757                                    ;ERROR  21      OPI SET WITH SECTOR PULSE,SECTOR COMPARE WAS RESET
1758
1759  001732  032300                            EMT21
1760  001734  037614                            EHT1
1761  001736  037724                            EDT1
1762  001740  037760                            EFT1
1763
1764
1765                                    ;ERROR  22      DTE SET WITH INDEX PULSE,SEARCH WAS NOT ENABLED
1766
1767  001742  032322                            EMT22
1768  001744  037614                            EHT1
1769  001746  037724                            EDT1
1770  001750  037760                            EFT1
1771
1772
1773                                    ;ERROR  23      DTE SET WITH SECTOR PULSE,SECTOR COMPARE WAS RESET
1774
1775  001752  032346                            EMT23
1776  001754  037614                            EHT1
1777  001756  037724                            EDT1
1778  001760  037760                            EFT1
1779
1780
1781                                    ;ERROR  24      DTE DID NOT SET WITH SECTOR PULSE,SECTOR COMPARE WAS SET
1782
1783  001762  032372                            EMT24
1784  001764  037614                            EHT1
1785  001766  037724                            EDT1
1786  001770  037760                            EFT1
1787
1788
1789                                    ;ERROR  25      DTF SET WITH SECTOR PULSE DURING FORMAT CHANGE
1790
1791  001772  032416                            EMT25
1792  001774  037614                            EHT1
1793  001776  037724                            EDT1
1794  002000  037760                            EFT1
```

```
1795
1796
1797                                    ;ERROR  26      MBA CLR L IS STUCK ACTIVE
1798
1799  002002  032436                            EMT26
1800  002004  000000                            0
1801  002006  000000                            0
1802  002010  000000                            0
1803
1804
1805                                    ;ERROR  27      COULD NOT SET DTE AFTER FORMAT CHANGE
1806
1807  002012  032470                            EMT27
1808  002014  037614                            EHT1
1809  002016  037724                            EDT1
1810  002020  037760                            EFT1
1811
1812
1813                                    ;ERROR  30      CANNOT SET PROM STROBE WITH BIT CLOCK
1814
1815  002022  032510                            EMT30
1816  002024  037614                            EHT1
1817  002026  037724                            EDT1
1818  002030  037760                            EFT1
1819
1820
1821                                    ;ERROR  31      CANNOT CLEAR RMER1,DTE
1822
1823  002032  032530                            EMT31
1824  002034  037614                            EHT1
1825  002036  037724                            EDT1
1826  002040  037760                            EFT1
1827
1828
1829                                    ;ERROR  32      PROM STROBE RESET EARLY
1830
1831  002042  032544                            EMT32
1832  002044  037614                            EHT1
1833  002046  037724                            EDT1
1834  002050  037760                            EFT1
1835
1836
1837                                    ;ERROR  33      PROM STROBE SET EARLY
1838
1839  002052  032556                            EMT33
1840  002054  037614                            EHT1
1841  002056  037724                            EDT1
1842  002060  037760                            EFT1
1843
1844
1845                                    ;ERROR  34      LOOKING FOR SYNC SET EARLY
1846
1847  002062  032570                            EMT34
1848  002064  037614                            EHT1
1849  002066  037724                            EDT1
1850  002070  037760                            EFT1
```

```
1851
1852
1853                                    ;ERROR  35      LOOKING FOR SYNC DID NOT SET
1854
1855  002072  032602                            EMT35
1856  002074  037614                            EHT1
1857  002076  037724                            EDT1
1858  002100  037760                            EFT1
1859
1860
1861                                    ;ERROR  36      PROM STROBE  SET WHILE LOOKING FOR SYNC
1862
1863  002102  032614                            EMT36
1864  002104  037614                            EHT1
1865  002106  037724                            EDT1
1866  002110  037760                            EFT1
1867
1868
1869                                    ;ERROR  37      SYNC DETECTED WITH WRONG PATTERN
1870
1871  002112  032634                            EMT37
1872  002114  037670                            EHT115
1873  002116  037750                            EDT115
1874  002120  037776                            EFT115
1875
1876
1877                                    ;ERROR  40      SYNC NOT DETECTED
1878
1879  002122  032664                            EMT40
1880  002124  037614                            EHT1
1881  002126  037724                            EDT1
1882  002130  037760                            EFT1
1883
1884
1885                                    ;ERROR  41      DRIVE TIMING ERROR DID NOT CLEAR LOOKING FOR SYNC
1886
1887  002132  032706                            EMT41
1888  002134  037614                            EHT1
1889  002136  037724                            EDT1
1890  002140  037760                            EFT1
1891
1892
1893                                    ;ERROR  42      WRITE GATE DID NOT COME ON OR RESET EARLY
1894
1895  002142  032734                            EMT42
1896  002144  037614                            EHT1
1897  002146  037724                            EDT1
1898  002150  037760                            EFT1
1899
1900
1901                                    ;ERROR  43      INCORRECT SYNC PATTERN DURING HEADER
1902
1903  002152  032752                            EMT43
1904  002154  000000                            0
1905  002156  000000                            0
1906  002160  000000                            0
```

```
1907
1908
1909                                    ;ERROR  44      INCORRECT SYNC PATTERN DURING HEADER
1910
1911  002162  032752                              EMT43
1912  002164  037630                              EHT7
1913  002166  037730                              EDT5
1914  002170  037764                              EFT5
1915
1916
1917                                    ;ERROR  45      HEADER AREA DID NOT COME ON OR RESET EARLY
1918
1919  002172  033004                              EMT45
1920  002174  037614                              EHT1
1921  002176  037724                              EDT1
1922  002200  037760                              EFT1
1923
1924
1925                                    ;ERROR  46      CRC ENABLE DID NOT SET
1926
1927  002202  033020                              EMT46
1928  002204  037614                              EHT1
1929  002206  037724                              EDT1
1930  002210  037760                              EFT1
1931
1932
1933                                    ;ERROR  47      INCORRECT HEADER GENERATED DURING WRITE
1934
1935  002212  033034                              EMT47
1936  002214  037634                              EHT47
1937  002216  037732                              EDT47
1938  002220  037760                              EFT1
1939
1940
1941                                    ;ERROR  50      READ GATE INCORRECT DURING HEADER AREA
1942
1943  002222  033052                              EMT50
1944  002224  037614                              EHT1
1945  002226  037724                              EDT1
1946  002230  037760                              EFT1
1947
1948
1949                                    ;ERROR  51      UNEXPECTED HEADER ERROR DURING DIAGNOSTIC MODE
1950
1951  002232  033154                              EMT55
1952  002234  037614                              EHT1
1953  002236  037724                              EDT1
1954  002240  037760                              EFT1
1955
1956
1957                                    ;ERROR  52      INCORRECT HEADER READ IN DIAGNOSTIC MODE
1958
1959  002242  033104                              EMT52
1960  002244  037640                              EHT52
1961  002246  037734                              EDT52
1962  002250  037766                              EFT57
```

```
1963
1964
1965                                    ;ERROR  53      INCORRECT TAG BUS DURING DATA COMMAND
1966
1967  002252  037510                              EMT276
1968  002254  037614                              EHT1
1969  002256  037724                              EDT1
1970  002260  037760                              EFT1
1971
1972
1973                                    ;ERROR  54      DATA TIMING SEQUENCER CONTROLS INCORRECT DURING DATA COMMAND
1974
1975  002262  033136                              EMT54
1976  002264  037614                              EHT1
1977  002266  037724                              EDT1
1978  002270  037760                              EFT1
1979
1980
1981                                    ;ERROR  55      DATA AREA DID NOT COME ON OR RFSET EARLY
1982
1983  002272  033154                              EMT55
1984  002274  037614                              EHT1
1985  002276  037724                              EDT1
1986  002300  037760                              EFT1
1987
1988
1989                                    ;ERROR  56      ECC ENABLE DID NOT SET
1990
1991  002302  033172                              EMT56
1992  002304  037614                              EHT1
1993  002306  037724                              EDT1
1994  002310  037760                              EFT1
1995
1996
1997                                    ;ERROR  57      DEVICE IS NOT AN RM03
1998
1999  002312  033206                              EMT57
2000  002314  037644                              EHT57
2001  002316  037736                              EDT57
2002  002320  037766                              EFT57
2003
2004
2005                                    ;ERROR  60      DEVICE AVAILABLE IS NOT SET
2006
2007  002322  033222                              EMT60
2008  002324  037614                              EHT1
2009  002326  037724                              EDT1
2010  002330  037760                              EFT1
2011
2012
2013                                    ;ERROR  61      INCORRECT ECC PATTERN GENERATED DURING ▲RITE
2014
2015  002332  033236                              EMT61
2016  002334  037650                              EHT61
2017  002336  037740                              EDT61
2018  002340  037766                              EFT57
```

```
2019
2020
2021                                    ;ERROR  62      CANNOT CLEAR PROM STROBE WITH BIT CLOCK
2022
2023  002342  033254                            EMT62
2024  002344  037614                            EHT1
2025  002346  037724                            EDT1
2026  002350  037760                            EFT1
2027
2028
2029                                    ;ERROR  63      DATA AREA DID NOT RESET
2030
2031  002352  033272                            EMT63
2032  002354  037614                            EHT1
2033  002356  037724                            EDT1
2034  002360  037760                            EFT1
2035
2036
2037                                    ;ERROR  64      UNEXPECTED ECC ERROR IN DIAGNOSTIC MODE
2038
2039  002362  033304                            EMT64
2040  002364  037614                            EHT1
2041  002366  037724                            EDT1
2042  002370  037760                            EFT1
2043
2044
2045                                    ;ERROR  65      INCORRECT DATA TRANSFERRED TO MEMORY
2046
2047  002372  033320                            EMT65
2048  002374  037614                            EHT1
2049  002376  037724                            EDT1
2050  002400  037760                            EFT1
2051
2052                                    ;ERROR  66
2053
2054  002402  033332                            EMT66
2055  002404  000000                            0
2056  002406  000000                            0
2057  002410  000000                            0
2058
2059                                    ;ERROR 67
2060
2061  002412  033346                            EMT67
2062  002414  000000                            0
2063  002416  000000                            0
2064  002420  000000                            0
2065
2066                                    ;ERROR 70
2067
2068  002422  033364                            EMT70
2069  002424  000000                            0
2070  002426  000000                            0
2071  002430  000000                            0
2072
2073                                    ;ERROR 71
2074
```

```
 2075  002432  033404                    EMT71
 2076  002434  000000                    0
 2077  002436  000000                    0
 2078  002440  000000                    0
 2079
 2080                              ;ERROR 72
 2081
 2082  002442  033430                    EMT72
 2083  002444  000000                    0
 2084  002446  000000                    0
 2085  002450  000000                    0
 2086
 2087                              ;ERROR 73
 2088
 2089  002452  033454                    EMT73
 2090  002454  000000                    0
 2091  002456  000000                    0
 2092  002460  000000                    0
 2093
 2094                              ;ERROR 74
 2095
 2096  002462  033474                    EMT74
 2097  002464  000000                    0
 2098  002466  000000                    0
 2099  002470  000000                    0
 2100
 2101                              ;ERROR 75
 2102
 2103  002472  033504                    EMT75
 2104  002474  000000                    0
 2105  002476  000000                    0
 2106  002500  000000                    0
 2107
 2108                              ;ERROR 76
 2109
 2110  002502  033516                    EMT76
 2111  002504  000000                    0
 2112  002506  000000                    0
 2113  002510  000000                    0
 2114
 2115                              ;ERROR 77
 2116
 2117  002512  033532                    EMT77
 2118  002514  000000                    0
 2119  002516  000000                    0
 2120  002520  000000                    0
 2121
 2122
 2123                              ;ERROR  100     CANT SET VOLUME VALID
 2124
 2125  002522  035340                    EMT170
 2126  002524  037710                    EHT150
 2127  002526  037750                    EDT115
 2128  002530  037776                    EFT115
 2129
 2130
```

```
2131                                    ;ERROR  101     RUN AND GO WONT SET
2132
2133  002532  037470                              EMT275
2134  002534  037614                              EHT1
2135  002536  037724                              EDT1
2136  002540  037760                              EFT1
2137
2138
2139                                    ;ERROR  102     SEARCH ENABLE WONT SET
2140
2141  002542  035366                              EMT172
2142  002544  037614                              EHT1
2143  002546  037724                              EDT1
2144  002550  037760                              EFT1
2145
2146
2147                                    ;ERROR  103     OCCUPIED IS INCORRECT FOR FUNCTION CODE
2148
2149  002552  035404                              EMT173
2150  002554  037710                              EHT150
2151  002556  037750                              EDT115
2152  002560  037776                              EFT115
2153
2154
2155                                    ;ERROR  104     READ IN PRESET DIDNT CLEAR RMDA, RMDC OR RMOF
2156
2157  002562  035426                              EMT174
2158  002564  000000                              0
2159  002566  000000                              0
2160  002570  000000                              0
2161
2162
2163                                    ;ERROR  105     READ IN PRESET DIDNT CLEAR RMOF
2164
2165  002572  035454                              EMT175
2166  002574  037614                              EHT1
2167  002576  037724                              EDT1
2168  002600  037760                              EFT1
2169
2170
2171                                    ;ERROR  106     READ IN PRESET DIDNT CLEAR RMDA
2172
2173  002602  035472                              EMT176
2174  002604  037614                              EHT1
2175  002606  037724                              EDT1
2176  002610  037760                              EFT1
2177
2178
2179                                    ;ERROR  107     READ IN PRESET DiDNT CLEAR RMDC
2180
2181  002612  035510                              EMT177
2182  002614  037614                              EHT1
2183  002616  037724                              EDT1
2184  002620  037760                              EFT1
2185
2186
```

```
 2187                                   ;ERROR  110     CANT SET OFFSET MODE BY OFFSET COMMAND
 2188
 2189  002622  035526                           EMT200
 2190  002624  037614                           EHT1
 2191  002626  037724                           EDT1
 2192  002630  037760                           EFT1
 2193
 2194
 2195
 2196
 2197                                   ;PUT ERROR TABLE HERE
 2198
```

```
2199                                    .SBTTL  START OF PROGRAM
2200
2201
2202  002632                            START:
2203
2204                                    ;CLEAR AND SETUP FOR TEST EXECUTION
2205  002632  000240                            NOP
2206  002634  000005                            RESET                   ;INITIALIZE THE SYSTEM
2207
2208                                    .SBTTL  INITIALIZE THE COMMON TAGS
2209                                    ;;CLEAR THE COMMON TAGS ($CMTAG) AREA
2210  002636  012706  001114                    MOV     #$CMTAG,R6      ;;FIRST LOCATION TO BE CLEARED
2211  002642  005026                            CLR     (R6)+           ;;CLEAR MEMORY LOCATION
2212  002644  022706  001154                    CMP     #SWR,R6 ;;DONE?
2213  002650  001374                            BNE     .-6             ;;LOOP BACK IF NO
2214  002652  012706  001100                    MOV     #STACK,SP       ;;SETUP THE STACK POINTER
2215                                    ;;INITIALIZE A FEW VECTORS
2216  002656  012737  026042  000020            MOV     #$SCOPE,@#IOTVEC ;;IOT VECTOR FOR SCOPE ROUTINE
2217  002664  012737  000340  000022            MOV     #340,@#IOTVEC+2 ;;LEVEL 7
2218  002672  012737  026424  000030            MOV     #$ERROR,@#EMTVEC ;;EMT VECTOR FOR ERROR ROUTINE
2219  002700  012737  000340  000032            MOV     #340,@#EMTVEC+2 ;;LEVEL 7
2220  002706  012737  030164  000034            MOV     #$TRAP,@#TRAPVEC ;;TRAP VECTOR FOR TRAP CALLS
2221  002714  012737  000340  000036            MOV     #340,@#TRAPVEC+2;LEVEL 7
2222  002722  012737  030272  000024            MOV     #$PWRDN,@#PWRVEC ;;POWER FAILURE VECTOR
2223  002730  012737  000340  000026            MOV     #340,@#PWRVEC+2 ;;LEVEL 7
2224  002736  013737  021506  021500            MOV     $ENDCT,$EOPCT   ;;SETUP END-OF-PROGRAM COUNTER
2225  002744  005037  001206                    CLR     $TIMES          ;;INITIALIZE NUMBER OF ITERATIONS
2226  002750  005037  001210                    CLR     $ESCAPE         ;;CLEAR THE ESCAPE ON ERROR ADDRESS
2227  002754  112737  000001  001131            MOVB    #1,$ERMAX       ;;ALLOW ONE ERROR PER TEST
2228  002762  012737  002762  001122            MOV     #.,$LPADR       ;;INITIALIZE THE LOOP ADDRESS FOR SCOPE
2229  002770  012737  002770  001124            MOV     #.,$LPERR       ;;SETUP THE ERROR LOOP ADDRESS
2230                                    ;;SIZE FOR A HARDWARE SWITCH REGISTER. IF NOT FOUND OR IT IS
2231                                    ;;EQUAL TO A "-1", SETUP FOR A SOFTWARE SWITCH REGISTER.
2232  002776  013746  000004                    MOV     @#ERRVEC,-(SP)  ;;SAVE ERROR VECTOR
2233  003002  012737  003036  000004            MOV     #64$,@#ERRVEC   ;;SET UP ERROR VECTOR
2234  003010  012737  177570  001154            MOV     #DSWR,SWR       ;;SETUP FOR A HARDWARE SWICH REGISTER
2235  003016  012737  177570  001156            MOV     #DDISP,DISPLAY  ;;AND A HARDWARE DISPLAY REGISTER
2236  003024  022777  177777  176122            CMP     #-1,@SWR        ;;TRY TO REFERENCE HARDWARE SWR
2237  003032  001012                            BNE     66$             ;;BRANCH IF NO TIMEOUT TRAP OCCURRED
2238                                                                    ;;AND  THE HARDWARE SWR IS NOT = -1
2239  003034  000403                            BR      65$             ;;BRANCH IF NO TIMEOUT
2240  003036  012716  003044            64$:    MOV     #65$,(SP)       ;;SET UP FOR TRAP RETURN
2241  003042  000002                            RTI
2242  003044  012737  000176  001154    65$:    MOV     #SWREG,SWR      ;;POINT TO SOFTWARE SWR
2243  003052  012737  000174  001156            MOV     #DISPREG,DISPLAY
2244  003060  012637  000004            66$:    MOV     (SP)+,@#ERRVEC  ;;RESTORE ERROR VECTOR
2245
2246  003064  005037  001230                    CLR     $PASS           ;;CLEAR PASS COUNT
2247  003070  132737  000200  001243            BITB    #APTSIZE,$ENVM  ;;TEST USER SIZE UNDER APT
2248  003076  001403                            BEQ     67$             ;;YES,USE NON-APT SWITCH
2249  003100  012737  001244  001154            MOV     #$SWREG,SWR     ;;NO,USE APT SWITCH REGISTER
2250  003106                            67$:
2251  003106  012746  000140                    MOV     #PR3,-(SP)      ;;PUT NEW PS ON STACK
2252  003112  012746  003120                    MOV     #68$,-(SP)      ;;PUT NEW PC ON  STACK
2253  003116  000002                            RTI                     ;;POP NEW PC AND PS
2254  003120                            68$:
```

```
2255                                    .SBTTL  TYPE PROGRAM NAME
2256                                    ;;TYPE THE NAME OF THE PROGRAM IF FIRST PASS
2257  003120  005227  177777                    INC     #-1             ;;FIRST TIME?
2258  003124  001061                          BNE     69$             ;;BRANCH IF NO
2259  003126  022737  021642  000042          CMP     #$ENDAD,@#42    ;;ACT-11?
2260  003134  001455                          BEQ     69$             ;;BRANCH IF YES
2261  003136  104401  003204                TYPE    ,70$            ;;TYPE ASCIZ STRING
2262                                    .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
2263  003142  005737  000042                TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
2264  003146  001012                          BNE     71$             ;;BRANCH IF YES
2265  003150  123727  001242  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
2266  003156  001406                          BEQ     71$             ;;BRANCH IF YES
2267  003160  023727  001154  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
2268  003166  001005                          BNE     72$             ;;BRANCH IF NO
2269  003170  104407                          GTSWR                   ;;GET SOFT-SWR SETTINGS
2270  003172  000403                          BR      72$
2271  003174  112737  000001  001150  71$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
2272  003202                          72$:
2273  003202  000432                          BR      69$             ;;GET OVER THE ASCIZ
2274                                  ;;70$:  .ASCIZ  <CRLF>@CZRMKA0, RM03/RM02 DISKLESS DIAGNOSTIC  PART II @<CRLF>
2275  003270                          69$.
2276
2277
2278                                    ;FIND OUT IF PROGRAM IS RUNNING IN STANDALONE MODE
2279  003270  005737  000042                TST     42              ;IS LOC 42 ZERO ??
2280  003274  001003                          BNE     10$             ;NO - NOT IN STANDALONE
2281  003276  105737  001242                TSTB    $ENV            ;IS APT ENVIRONMENT ZERO ??
2282  003302  001451                          BEQ     STANDALONE      ;YES - PROGRAM IN STANDALONE
2283  003304                          10$:
2284
2285                                    ;PROGRAM NOT RUNNING IN STANDALONE - SEE IF SIZING IS ALLOWED
2286  003304  132737  000200  001243  XSIZ:   BITB    #BIT7,$ENVM     ;SIZING ALLOWED ??
2287  003312  001043                          BNE     20$             ;NO
2288                                  ;       MOV     #377,$DEVM      ;YES - SET DEVICE MAP FOR ALL DEVICES
2289  003314  005037  001300                CLR     $DEVM           ;CLEAR THE BIT MAP
2290  003320  005001                          CLR     R1              ;START FROM THE DRIVE 0
2291  003322  012704  000001                MOV     #BIT0,R4        ;BIT MAP DOR DRIVE 0
2292  003326  013700  001276                MOV     $BASE,R0                ;BASE ADDRESS
2293  003332  012760  000040  000010  15$:   MOV     #CLR,RMCS2(R0)  ;LOAD THE DRIVE ADDRESS
2294  003340  010160  000010                MOV     R1,RMCS2(R0)    ;LOAD THE DRIVE ADDRESS
2295  003344  016003  000010                MOV     RMCS2(R0),R3    ;NED BIT SET ?
2296  003350  032703  010000                BIT     #NED,R3         ;BRANCH IF SO
2297  003354  001010                          BNE     16$             ;TO NEXT DRIVE
2298  003356  016003  000000                MOV     RMCS1(R0),R3    ;CHECK THE DVA BIT
2299  003362  032703  004000                BIT     #DVA,R3         ;DRIVE AVAILABLE ?
2300  003366  001403                          BEQ     16$             ;BRANCH IF DRIVE NOT AVAILABLE
2301  003370  050437  001300                BIS     R4,$DEVM        ;SET THE BIT MAP FOR ALL AVAILABLE DRIVE
2302  003374  000405                          BR      17$             ;DONT TYPE THE NONE EXIST MESG
2303  003376                          16$:
2304  003376  104401  031553                TYPE    ,NOTEX          ;TYPE NOT EXIT MESSAGE
2305  003402  010146                          MOV     R1,-(SP)        ;DRIVE NUMBER
2306  003404  104403                          TYPOS
2307  003406     006                          .BYTE   6
2308  003407     000                          .BYTE   0
2309  003410  005201                  17$:    INC     R1              ;INCREMENT THE DRIVE ADDRESS
2310  003412  006304                          ASL     R4              ;SET UP THE BIT MAP FOR NEXT DRIVE
```

```
 2311  003414  022701  000007                  CMP     #7,R1           ;ALL DRIVE ARE CHECKED ?
 2312  003420  103344                          BHIS    15$             ;BRANCH IF NOT
 2313  003422                          20$:
 2314
 2315                                  ;GO TO COMMON START CODE
 2316  003422  000137  004254                  JMP     CMNSTART
```

```
2317  003426                          STANDALONE:
2318
2319  003426  004737  026634                  JSR     PC,$TKINT
2320                                  ;SEE IF THIS IS THE FIRST START AFTER PROGRAM WAS LOADED
2321  003432  005327  000001                  DEC     #1              ;FIRST START ??
2322  003436  100024                          BPL     10$             ;YES !!
2323
2324
2325                                  ;SEE IF THE USER WANTS TO KEEP SAME DEVICES FOR TESTING
2326  003440  104401  031062          5$:     TYPE    ,CNSL00         ;MAINTAIN PREVIOUS PARAMETERS??
2327  003444  104411                          RDCHR                   ;GET RESPONSE
2328  003446  012637  001176                  MOV     (SP)+,$TMP1     ;ECHO RESPONSE
2329  003452  104401  001176                  TYPE    ,$TMP1
2330  003456  123727  001176  000131        CMPB    $TMP1,#'Y       ;YES RESPONSE??
2331  003464  001002                          BNE     6$              ;NO!!
2332  003466  000137  004414                  JMP     READY           ;KEEP PREVIOUS PARAMETERS
2333  003472  123727  001176  000116  6$:     CMPB    $TMP1,#'N       ;NO RESPONSE??
2334  003500  001420                          BEQ     20$             ;GET NEW PARAMETERS
2335  003502  104401  030725                  TYPE    ,QSTMRK         ;NOT YES OR NO, TYPE "?"
2336  003506  000754                          BR      5$              ;RETRY
2337  003510                          10$:
2338
2339                                  ;SEE IF OPERATOR WANTS HELP FILE
2340  003510  104401  030727                  TYPE    ,HELPQST        ;WANT HELP ??
2341  003514  104411                          RDCHR                   ;GET RESPONSE
2342  003516  012637  001176                  MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
2343  003522  104401  001176                  TYPE    ,$TMP1
2344  003526  123727  001176  000131        CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ??
2345  003534  001002                          BNE     20$             ;NO - DONT TYPE HELP
2346  003536  104401  054062                  TYPE    ,HELP           ;YES - TYPE HELP TEXT
2347  003542                          20$:
2348
2349                                  ;SEE IF USER WANTS TO CHANGE RM03 UNIBUS ADDRESS
2350  003542  104401  030763                  TYPE    ,UBUSQST        ;WANT TO CHANGE ADDRESS ??
2351  003546  104411                          RDCHR                   ;GET RESPONSE
2352  003550  012637  001176                  MOV     (SP)+,$TMP1     ;SAVE AND ECHO RESPONSE
2353  003554  104401  001176                  TYPE    ,$TMP1
2354  003560  104411                          RDCHR                   ;READ S CR
2355  003562  005726                          TST     (SP)+           ;CLEAR THE STACK
2356  003564  123727  001176  000131        CMPB    $TMP1,#'Y       ;WAS IT A YES RESPONSE ??
2357  003572  001137                          BNE     90$             ;NO !!
2358  003574                          30$:
2359
2360                                  ;DIALOGUE TO CHANGE THE UNIBUS ADDRESS, AND INTERRUPT VECTOR
2361  003574  104401  031121                  TYPE    ,CNSL01         ;TYPE CURRENT BUS ADDRESS
2362  003600  013746  001276                  MOV     $BASE,-(SP)     ;;SAVE $BASE FOR TYPEOUT
2363  003604  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
2364  003606  104401  030716                  TYPE    ,CLSPRN
2365  003612  104413                          RDOCT                   ;GET NEW BUS ADDRESS
2366  003614  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN??
2367  003620  001412                          BEQ     50$             ;YES-SKIP TO NEXT ENTRY
2368  003622  022737  160000  001176        CMP     #160000,$TMP1   ;BASE ADDRESS IN I/O PAGE??
2369  003630  101403                          BLOS    40$             ;YES
2370  003632  104401  031146                  TYPE    ,CNSL02         ;TYPE WARNING MESSAGE
2371  003636  000756                          BR      30$             ;RETRY
2372  003640  013737  001176  001276  40$:     MOV     $TMP1,$BASE     ;STORE NEW BUS ADDRESS
```

```
2373  003646  113737  001272  001176  50$:    MOVB    $VECT1,$TMP1    ;TYPE CURRENT VECTOR ADDRESS
2374  003654  105037  001177                  CLRB    $TMP1+1
2375  003660  104401  031227                  TYPE    ,CNSL03
2376  003664  013746  001176                  MOV     $TMP1,-(SP)     ;;SAVE $TMP1 FOR TYPEOUT
2377  003670  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2378  003672     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
2379  003673     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2380  003674  104401  030716                  TYPE    ,CLSPRN
2381  003700  104413                          RDOCT                   ;GET NEW VECTOR ADDRESS
2382  003702  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN?
2383  003706  001412                          BEQ     70$             ;YES-SKIP TO NEXT ENTRY
2384  003710  022737  001000  001176          CMP     #1000,$TMP1     ;VECTOR ADDRESS < 1000??
2385  003716  101003                          BHI     60$             ;YES!!
2386  003720  104401  031257                  TYPE    ,CNSL04         ;TYPE WARNING MESSAGE
2387  003724  000750                          BR      50$             ;RETRY
2388  003726  113737  001176  001272  60$:    MOVB    $TMP1,$VECT1    ;STORE NEW VECTOR ADDRESS
2389  003734  113737  001273  001176  70$:    MOVB    $VECT1+1,$TMP1  ;TYPE CURRENT PRIORITY
2390  003742  006237  001176                  ASR     $TMP1
2391  003746  006237  001176                  ASR     $TMP1
2392  003752  006237  001176                  ASR     $TMP1
2393  003756  006237  001176                  ASR     $TMP1
2394  003762  006237  001176                  ASR     $TMP1
2395  003766  105037  001177                  CLRB    $TMP1+1
2396  003772  104401  031333                  TYPE    ,CNSL05
2397  003776  013746  001176                  MOV     $TMP1,-(SP)     ;;SAVE $TMP1 FOR TYPEOUT
2398  004002  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
2399  004004     001                          .BYTE   1               ;;TYPE 1 DIGIT(S)
2400  004005     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
2401  004006  104401  030716                  TYPE    ,CLSPRN
2402  004012  104413                          RDOCT                   ;GET NEW PRIORITY
2403  004014  012637  001176                  MOV     (SP)+,$TMP1     ;CARRIAGE RETURN??
2404  004020  001424                          BEQ     90$             ;YES-SKIP TO NEXT ENTRY
2405  004022  023727  001176  000007          CMP     $TMP1,#7        ;LEGAL PRIORITY??
2406  004030  002403                          BLT     80$             ;YES!!
2407  004032  104401  031367                  TYPE    ,CNSL06         ;TYPE WARNING MESSAGE
2408  004036  000736                          BR      70$             ;RETRY
2409  004040                          80$:                            ;STORE NEW PRIORITY
2410  004040  006337  001176                  ASL     $TMP1
2411  004044  006337  001176                  ASL     $TMP1
2412  004050  006337  001176                  ASL     $TMP1
2413  004054  006337  001176                  ASL     $TMP1
2414  004060  006337  001176                  ASL     $TMP1
2415  004064  113737  001176  001273          MOVB    $TMP1,$VECT1+1
2416  004072                          90$:
2417
2418                                  ;DIALOGUE TO INPUT DEVICE NUMBERS
2419  004072  005037  001300                  CLR     $DEVM           ;CLEAR DEVICE MAP
2420  004076  104401  031414                  TYPE    ,CNSL07         ;TYPE INPUT INSTRUCTIONS
2421  004102  104401  030721                  TYPE    ,PROMPT         ;TYPE PROMPTING CHARACTER
2422  004106  104411                          RDCHR                   ;GET RESPONSE
2423  004110  012637  001176                  MOV     (SP)+,$TMP1     ;ECHO RESPONSE
2424  004114  104401  001176                  TYPE    ,$TMP1
2425  004120  023727  001176  000101          CMP     $TMP1,#'A       ;TEST ALL DRIVES??
2426  004126  001021                          BNE     110$            ;NO
2427  004130  000137  003304                  JMP     XSIZ            ;TO SIZE ALL POWER UP DRIVES
2428  004134  012737  000377  001300          MOV     #377,$DEVM      ;TEST ALL DEVICES
```

```
2429  004142  000444                          BR      140$            ;SKIP TO NEXT ENTRY
2430  004144  104401  030721          100$:   TYPE    ,PROMPT         ;TYPE PROMPTING CHARACTER
2431  004150  104411                          RDCHR                   ;GET RESPONSE
2432  004152  012637  001176                  MOV     (SP)+,$TMP1     ;ECHO RESPONSE
2433  004156  104401  001176                  TYPE    ,$TMP1
2434  004162  023727  001176  000015          CMP     $TMP1,#CR       ;CARRIAGE RETURN??
2435  004170  001431                          BEQ     140$
2436  004172  023727  001176  000060  110$:   CMP     $TMP1,#'0       ;NUMBER < 0??
2437  004200  002404                          BLT     120$            ;YES!!
2438  004202  023727  001176  000067          CMP     $TMP1,#'7       ;NUMBER > 7??
2439  004210  003403                          BLE     130$            ;NO!!
2440  004212  104401  030725          120$:   TYPE    ,QSTMRK         ;TYPE "?"
2441  004216  000752                          BR      100$            ;RETRY
2442  004220  013701  001176          130$:   MOV     $TMP1,R1        ;R1=DRIVE NUMBER
2443  004224  042701  177770                  BIC     # C7,R1
2444  004230  116102  031700                  MOVB    ATNTBL(R1),R2           ;DECODE DEVICE NUMBER
2445  004234  042702  177400                  BIC     # C377,R2              ;CLEAR UNUSED BITS
2446  004240  050237  001300                  BIS     R2,$DEVM                ;SET DEVICE # IN MAP
2447  004244  122737  000377  001300          CMPB    #377,$DEVM      ;DONE ??
2448  004252  101334                          BHI     100$            ;NO
2449  004254                          140$:
2450
```

```
 2451  004254                          CMNSTART:
 2452
 2453                                  ;ASSEMBLE TEST QUE FROM DEVICE MAP
 2454  004254  013700  001300                  MOV     $DEVM,R0        ;R0 = DEVICE MAP
 2455  004260  012701  001460                  MOV     #TSTQUE+2,R1    ;R1 = ADDRESS OF FIRST ENTRY IN QUE
 2456  004264  010137  001456                  MOV     R1,TSTQUE       ;INITIALIZE ENTRY POINTER
 2457  004270  012702  000001                  MOV     #1,R2           ;R2 = DEVICE POINTER
 2458  004274  005003                          CLR     R3              ;R3 = DEVICE NUMBER
 2459  004276  030200                  10$:    BIT     R2,R0           ;IS THIS DEVICE IN MAP ??
 2460  004300  001406                          BEQ     20$             ;NO !!
 2461  004302  010311                          MOV     R3,(R1)         ;YES - ENTER DEVICE NUMBER IN QUE
 2462  004304  116361  031700  000001          MOVB    ATNTBL(R3),1(R1);ENTER ATTENTION BIT IN QUE
 2463  004312  062701  000002                  ADD     #2,R1           ;ADVANCE ENTRY POINTER
 2464  004316  006302                  20$:    ASL     R2              ;ADVANCE DEVICE POINTER
 2465  004320  105702                          TSTB    R2              ;DONE ALL DEVICES ??
 2466  004322  001402                          BEQ     25$             ;YES
 2467  004324  005203                          INC     R3              ;ADVANCE DEVICE NUMBER
 2468  004326  000763                          BR      10$             ;ENTER NEXT DEVICE
 2469  004330  005011                  25$:    CLR     (R1)            ;TERMINATE TEST QUE
 2470
 2471                                  ;SIZE FOR CLOCK
 2472  004332  004737  022464                  JSR     PC,SIZCLK       ;SEE IF CLOCK PRESENT
 2473  004336  000403                          BR      40$             ;YES - CLOCK IS PRESENT
 2474  004340  104000                  30$:    ERROR                   ;NO CLOCK
 2475  004342  000000                          HALT
 2476  004344  000775                          BR      30$
 2477  004346                          40$:
 2478                                  .SBTTL  GET VALUE FOR SOFTWARE SWITCH REGISTER
 2479  004346  005737  000042                  TST     @#42            ;;ARE WE RUNNING UNDER XXDP/ACT?
 2480  004352  001012                          BNE     64$             ;;BRANCH IF YES
 2481  004354  123727  001242  000001          CMPB    $ENV,#1         ;;ARE WE RUNNING UNDER APT?
 2482  004362  001406                          BEQ     64$             ;;BRANCH IF YES
 2483  004364  023727  001154  000176          CMP     SWR,#SWREG      ;;SOFTWARE SWITCH REG SELECTED?
 2484  004372  001005                          BNE     65$             ;;BRANCH IF NO
 2485  004374  104407                          GTSWR                   ;;GET SOFT-SWR SETTINGS
 2486  004376  000403                          BR      65$
 2487  004400  112737  000001  001150  64$:    MOVB    #1,$AUTOB       ;;SET AUTO-MODE INDICATOR
 2488  004406                          65$:
 2489  004406  012737  000140  000032          MOV     #PR3,@#EMTVEC+2 ;DROP PRIORITY DURING ERROR TYPEOUT
 2490  004414  000240                  READY:  NOP                     ;READY TO START TEST
 2491  004416  004737  026634                  JSR     PC,$TKINT       ;INITIALIZE TTY
 2492  004422  117737  175030  001234          MOVB    @TSTQUE,$UNIT   ;LOAD UNIT NUMBER
```

```
                                .SBTTL  REGISTER AND STORAGE USAGE

                                ;REGISTER ASSIGNMENTS
```

```
2496                            ;
2497                            ;R0     =       UNIBUS ADDRESS OF RH CONTROLLER
2498                            ;R1     =       ADDRESS OF ENTRY IN TEST QUE CORRESPONDING TO THE
2499                            ;               UNIT UNDER TEST
2500                            ;R2,R3  =       WORKING REGISTERS FOR TEST IN PROGRESS, MUST BE
2501                            ;               SAVED BY SUBROUTINES
2502                            ;R4,R5  =       GENERAL WORKING REGISTERS, ARE NOT SAVED BY
2503                            ;               SUBROUTINES
2504                            ;R6     =       STACK POINTER
2505                            ;R7     =       LINKAGE REGISTER TO SUBROUTINES
2506
2507                            ;STORAGE ASSIGNMENTS
2508                            ;
2509                            ;$TMP0-$TMP4    TEMPORARY STORAGE, NOT SAVED BY SUBROUTINES
2510                            ;$GDDAT,$BDDAT  EXPECTED AND RRECEIVED STATUS FOR ERROR TYPEOUT
2511                            ;$GDADR,$BDADR  ADDRESS OF EXPECTED AND RECEIVED STATUS IF APPLICABLE,
2512                            ;               ALSO THE ADDRESS OF A REGISTER ERROR
2513                            ;
2514                            ;$TSTN  =       TEST NUMBER
2515                            ;$UNIT  =       NUMBER OF DEVICE BEING TESTED
2516                            ;RGINBF =       THE REGISTER INPUT BUFFER HAS A STORAGE LOCATION FOR
2517                            ;               EACH REGISTER, AND IS USED WHEN READING STATUS AND
2518                            ;               CONTROL DATA
2519                            ;RGOTBF =       THE REGISTER OUTPUT BUFFER HAS A STORAGE LOCATION FOR
2520                            ;               EACH REGISTER, AND IS USED FOR ASSEMBLING DATA TO BE
2521                            ;               WRITTEN IN REGISTERS
```

```
2522                                    ;;*****************************************************************
2523                                    ;*TEST 1        TRANSFER TEST
2524
2525                                    ;;*****************************************************************
2526  004430  000004                    TST1:   SCOPE
2527  004432  012737  000001  001226            MOV     #1,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2528
2529  004440  000240                            NOP
2530  004442  012737  000024  001120            MOV     #20.,$ICNT      ;20 ITERATIONS
2531  004450  112737  000001  001131            MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2532  004456  012737  004472  001122            MOV     #T1,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2533  004464  012737  004472  001124            MOV     #T1,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2534  004472                            T1:
2535  004472  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
2536  004476  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2537  004502  013701  001456                    MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2538  004506  012702  000000                    MOV     #0,R2           ;R2 = REGISTER INDEX
2539  004512                            10$:
2540                                    ;CLEAR THE MASSBUS AND VERIFY THAT NONEXISTANT DEVICE ERROR IS RESET
2541  004512  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2542  004520  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2543
2544  004524  016037  000010  001142            MOV     RMCS2(R0),$BDDAT        ;STORE RMCS2 AT $BDDAT
2545  004532  032737  010000  001142            BIT     #NED,$BDDAT
2546  004540  001417                            BEQ     20$
2547  004542  111137  001140                    MOVB    (R1),$GDDAT
2548  004546  042737  177770  001140            BIC     # CUNTMSK,$GDDAT
2549  004554  052737  000100  001140            BIS     #IR,$GDDAT
2550  004562  010037  001136                    MOV     R0,$BDADR
2551  004566  062737  000010  001136            ADD     #RMCS2,$BDADR
2552  004574  104001                            ERROR   1
2553  004576  000500                            BR      60$
2554  004600                            20$:
2555                                    ;READ THE REGISTER WHOSE INDEX IS IN R2 AND EXIT TEST IF THE READ
2556                                    ;DOES NOT SET "NED" ERROR
2557  004600  010003                            MOV     R0,R3           ;R3 = REGISTER ADDRESS
2558  004602  060203                            ADD     R2,R3
2559  004604  011304                            MOV     (R3), R4                ;READ REGISTER
2560  004606  032760  010000  000010            BIT     #NED, RMCS2(R0)         ;IS "NED" SET??
2561  004614  001473                            BEQ     70$             ;NO!!
2562  004616  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2563  004624  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2564
2565  004630  016037  000010  001142            MOV     RMCS2(R0),$BDDAT        ;STORE RMCS2 AT $BDDAT
2566  004636  032737  010000  001142            BIT     #NED,$BDDAT
2567  004644  001417                            BEQ     30$
2568  004646  111137  001140                    MOVB    (R1),$GDDAT
2569  004652  042737  177770  001140            BIC     # CUNTMSK,$GDDAT
2570  004660  052737  000100  001140            BIS     #IR,$GDDAT
2571  004666  010037  001136                    MOV     R0,$BDADR
2572  004672  062737  000010  001136            ADD     #RMCS2,$BDADR
2573  004700  104001                            ERROR   1
2574  004702  000436                            BR      60$
2575  004704                            30$:
2576                                    ;WRITE THE REGISTER WHOSE INDEX IS IN R2 AND EXIT TEST IF THE WRITE
2577                                    ;DOES NOT SET "NED" ERROR
```

```
2578  004704  012713  000000          MOV     #0,(R3)          ;WRITE REGISTER
2579  004710  032760  010000  000010  BIT     #NED, RMCS2      (R0)    ;IS "NED" SET??
2580  004716  001432                  BEQ     70$              ;NO!!
2581
2582  004720                  40$:
2583                          ;COULD NOT READ OR WRITE THE REGISTER WITHOUT SETING "NED" ERROR -
2584                          ;ADVANCE THE REGISTER INDEX AND REPEAT THE TEST FOR THE NEXT
2585                          ;AVAILABLE DEVICE REGISTER
2586  004720  062702  000002          ADD     #2,R2            ;ADVANCE TO NEXT REGISTER
2587  004724  022702  000002          CMP     #RMWC,R2                 ;IS THIS RMWC??
2588  004730  001773                  BEQ     40$              ;YES - TRY NEXT REGISTER
2589  004732  022702  000004          CMP     #RMBA,R2                 ;IS THIS RMBA??
2590  004736  001770                  BEQ     40$              ;YES - TRY NEXT REGISTER
2591  004740  022702  000010          CMP     #RMCS2,R2                ;IS THIS RMCS5??
2592  004744  001765                  BEQ     40$              ;YES - TRY ANOTHER REGISTER
2593  004746  022702  000016          CMP     #RMAS,R2                 ;IS THIS RMAS ??
2594  004752  001762                  BEQ     40$              ;YES - TRY ANOTHER REGISTER
2595  004754  022702  000022          CMP     #RMDB,R2                 ;IS THIS RMDB??
2596  004760  001757                  BEQ     40$              ;YES - TRY ANOTHER REGISTER
2597  004762  022702  000046          CMP     #RMEC2,R2                ;IS THIS A LEGAL REGISTER
2598  004766  103251                  BHIS    10$              ;YES - TRY THIS REGISTER
2599
2600  004770                  50$:
2601
2602                          ;GOT 'NONEXISTENT DEVICE' ERROR FOR EVERY REMOTE REGISTER ADDRESS
2603  004770  013737  001276  001136          MOV     $BASE, $BDADR            ;STORE BASE ADDRESS
2604  004776  104002                  ERROR   2                ;DEMAND OR TRANSFER FAILED
2605  005000  000137  021416          60$:    JMP     $EOSP            ;GO SELECT NEXT DEVICE
2606
2607  005004                  70$:
2608
2609                          ;;*****************************************************************
2610                          ;*TEST 2         CTOD TEST
2611
2612                          ;;*****************************************************************
2613  005004  000004          TST2:   SCOPE
2614  005006  012737  000002  001226          MOV     #2,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
2615  005014  000240                  NOP
2616  005016  012737  000024  001120          MOV     #20.,$ICNT       ;20 ITERATIONS
2617  005024  112737  000001  001131          MOVB    #1,$ERMAX        ;ONE ERROR ALLOWED
2618  005032  012737  005046  001122          MOV     #T2,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2619  005040  012737  005046  001124          MOV     #T2,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2620  005046                  T2:
2621  005046  012706  001100          MOV     #STACK,SP        ;LOAD THE STACK POINTER
2622  005052  013700  001276          MOV     $BASE,R0         ;R0 = UNIBUS ADDRESS OF UUT
2623  005056  013701  001456          MOV     TSTQUE,R1        ;R1 = POINTER TO DEVICE
2624  005062  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2625  005070  111160  000010          MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2626                          ;WRITE ONES IN REMOTE REGISTERS
2627
2628  005074  012760  000076  000000          MOV     #ILF76,RMCS1(R0)         ;LOAD RMCS1
2629
2630  005102  012760  177777  000006          MOV     #-1,RMDA(R0)     ;LOAD RMDA
2631
2632  005110  012760  001777  000034          MOV     #CYLMSK,RMDC(R0)         ;LOAD RMDC
2633
```

```
2634  005116  012760  016200  000032          MOV     #FMT16!ECI!HCI!OFD,RMOF(R0)     ;LOAD RMOF
2635                          ;READ REMOTE REGISTERS TWICE
2636  005124  012702  000001                  MOV     #1,R2
2637  005130                  10$:
2638
2639  005130  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
2640
2641  005136  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2642
2643  005144  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2644
2645  005152  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
2646  005160  005302                          DEC     R2
2647  005162  100362                          BPL     10$
2648                          ;SEE IF ANY ONE BITS CAME BACK
2649  005164  042737  177701  001326          BIC     # CILF76,RMCS1I ;IS RMCS1 0??
2650  005172  001014                          BNE     20$             ;NO!!
2651  005174  005737  001334                  TST     RMDAI           ;IS RMDA 0??
2652  005200  001011                          BNE     20$             ;NO!!
2653  005202  042737  176000  001362          BIC     #XNUDC,RMDCI    ;IS RMDC 0??
2654  005210  001005                          BNE     20$             ;NO!!
2655  005212  042737  161577  001360          BIC     #XNUOF,RMOFI    ;IS RMOF 0 ??
2656  005220  001001                          BNE     20$             ;NO!!
2657                          ;CANNOT READ ANY ONE BITS FROM REMOTE REGISTERS
2658  005222  104003                          ERROR   3               ;CTOD MUST BE STUCK
2659  005224                  20$:
2660
2661                          ;;*****************************************************************
2662                          ;*TEST 3        MASSBUS INITIALIZE TEST
2663
2664                          ;;*****************************************************************
2665  005224  000004          TST3:   SCOPE
2666  005226  012737  000003  001226          MOV     #3,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2667
2668  005234  000240                          NOP
2669  005236  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
2670  005244  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2671  005252  012737  005266  001122          MOV     #T3,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2672  005260  012737  005266  001124          MOV     #T3,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2673  005266                  T3:
2674  005266  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2675  005272  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2676  005276  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2677  005302  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2678  005310  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2679                          ;WRITE ONES IN SELECTED REGISTERS
2680
2681  005314  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
2682
2683  005322  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1
2684
2685  005330  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD RMER2
2686                          ;USING CONTROLLER CLEAR, IE., BIT 5 OF RMCS2, INITIALIZE THE MASSBUS
2687  005336  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2688  005344  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2689                          ;READ THE REGISTERS THAT WERE WRITTEN
```

```
2690
2691  005350  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
2692
2693  005356  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2694
2695  005364  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2696                        ;SEE IF ANY REGISTER BITS WERE CLEARED
2697  005372  052737  177701  001326          BIS     # CILF76,RMCS1I         ;SET ANY BIT NOT WRITTEN
2698  005400  052737  001567  001370          BIS     #XNUER2,RMER2I
2699  005406  022737  177777  001326          CMP     #-1,RMCS1I              ;ANY ZEROS IN RMCS1??
2700  005414  001011                          BNE     10$             ;YES!!
2701  005416  022737  177777  001342          CMP     #-1,RMER1I              ;ANY ZEROS IN RMERI??
2702  005424  001005                          BNE     10$             ;YES!!
2703  005426  022737  177777  001370          CMP     #-1,RMER2I              ;ANY ZEROS IN RMER2??
2704  005434  001001                          BNE     10$
2705                        ;NONE OF THE BITS WERE CLEARED
2706  005436  104004                          ERROR   4               ·MASSBUS INIT FAILED
2707  005440                10$:
2708
2709                        ;;*****************************************************************
2710                        ;*TEST 4         CLEAR STUCK ACTIVE TEST
2711
2712                        ;;*****************************************************************
2713  005440  000004        TST4:   SCOPE
2714  005442  012737  000004  001226          MOV     #4,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2715
2716  005450  000240                          NOP
2717  005452  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
2718  005460  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2719  005466  012737  005502  001122          MOV     #T4,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2720  005474  012737  005502  001124          MOV     #T4,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2721  005502                T4:
2722  005502  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2723  005506  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2724  005512  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2725  005516  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2726  005524  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2727                        ;WRITE ONES IN TEST REGISTERS
2728
2729  005530  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1
2730
2731  005536  012760  177777  000042          MOV     #-1,RMER2(R0)   ;LOAD RMER2
2732
2733  005544  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
2734                        ;READ TEST REGISTERS AND SEE IF ANY BITS ARE ON
2735
2736  005552  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2737
2738  005560  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2739
2740  005566  016037  000024  001352          MOV     RMMR1(R0),RMMR1I        ;STORE RMMR1 IN INPUT BUFFER
2741  005574  042737  040000  001342          BIC     #UNS,RMER1I     ;DONT ACCEPT UNSAFE
2742  005602  001011                          BNE     10$             ;BRANCH IF ANY OTHER BITS ON
2743  005604  042737  040200  001370          BIC     #SKI!DVC,RMER2I ;DONT ACCEPT SKI OR DVC
2744  005612  001005                          BNE     10$             ;BRANCH IF ANY OTHER BITS ON
2745  005614  032737  000001  001352          BIT     #DMD,RMMR1I     ;BRANCH IF DMD IS ON
```

```
2746  005622  001001                        BNE     10$
2747  005624  104026                        ERROR   26              ;MBA CLR IS STUCK ACTIVE
2748  005626                        10$:
2749
2750
2751                                ;;*****************************************************************
2752                                ;*TEST 5         TRISTATE TRANSFER TEST
2753
2754                                ;;*****************************************************************
2755  005626  000004                TST5:   SCOPE
2756  005630  012737  000005  001226        MOV     #5,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
2757  005636  000240                        NOP
2758  005640  012737  000024  001120        MOV     #20.,$ICNT      ;20 ITERATIONS
2759  005646  112737  000001  001131        MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2760  005654  012737  005670  001122        MOV     #T5,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2761  005662  012737  005670  001124        MOV     #T5,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2762  005670                        T5:
2763  005670  012706  001100                MOV     #STACK,SP       ;LOAD THE STACK POINTER
2764  005674  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2765  005700  013701  001456                MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2766  005704  005002                        CLR     R2              ;CLEAR ERROR FLAGS
2767  005706  012760  000040  000010        MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2768  005714  111160  000010                MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2769                                ;WRITE ONES IN SELECTED REGISTERS
2770
2771  005720  012760  000076  000000        MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
2772
2773  005726  012760  177777  000006        MOV     #-1,RMDA(R0)    ;LOAD RMDA
2774
2775  005734  012760  177777  000014        MOV     #-1,RMER1(R0)   ;LOAD RMER1
2776
2777  005742  012760  177777  000032        MOV     #-1,RMOF(R0)    ;LOAD RMOF
2778
2779  005750  012760  177777  000042        MOV     #-1,RMER2(R0)   ;LOAD RMER2
2780                                ;WRITE ZEROS IN SELECTED REGISTERS
2781
2782  005756  012760  000000  000000        MOV     #0,RMCS1(R0)    ;LOAD RMCS1
2783
2784  005764  012760  000000  000006        MOV     #0,RMDA(R0)     ;LOAD RMDA
2785
2786  005772  012760  000000  000014        MOV     #0,RMER1(R0)    ;LOAD RMER1
2787
2788  006000  012760  000000  000032        MOV     #0,RMOF(R0)     ;LOAD RMOF
2789
2790  006006  012760  000000  000034        MOV     #0,RMDC(R0)     ;LOAD RMDC
2791
2792  006014  012760  000000  000042        MOV     #0,RMER2(R0)    ;LOAD RMER2
2793                                ;READ BACK ALL REGISTERS
2794
2795  006022  016037  000000  001326        MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
2796
2797  006030  016037  000006  001334        MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2798
2799  006036  016037  000014  001342        MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2800
2801  006044  016037  000032  001360        MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
```

```
2802
2803  006052  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2804
2805  006060  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2806                        ;CHECK EACH REGISTER CONTENT FOR ZERO BITS WRITTEN & READ
2807  006066  012702  177777                  MOV     #-1,R2          ;ACCUMULATE ZEROS IN R2
2808  006072  052737  177701  001326          BIS     # CILF76,RMCS1I ;SET ALL BITS NOT WRITTEN
2809  006100  052737  161577  001360          BIS     #XNUOF,RMOFI
2810  006106  052737  176000  001362          BIS     #XNUDC,RMDCI
2811  006114  052737  001567  001370          BIS     #XNUER2,RMER2I
2812  006122  005137  001326                  COM     RMCS1I          ;COMPLEMENT REGISTER CONTENTS
2813  006126  005137  001334                  COM     RMDAI
2814  006132  005137  001342                  COM     RMER1I
2815  006136  005137  001360                  COM     RMOFI
2816  006142  005137  001362                  COM     RMDCI
2817  006146  005137  001370                  COM     RMER2I
2818  006152  043702  001326                  BIC     RMCS1I,R2               ;ACCUMULATE ALL ZERO BITS
2819  006156  043702  001334                  BIC     RMDAI,R2
2820  006162  043702  001342                  BIC     RMER1I,R2
2821  006166  043702  001360                  BIC     RMOFI,R2
2822  006172  043702  001362                  BIC     RMDCI,R2
2823  006176  043702  001370                  BIC     RMER2I,R2
2824  006202  001407                          BEQ     10$             ;BRANCH IF EACH BIT IS ZERO
2825                        ;ONE OR MORE BIT POSITIONS ARE NOT ZERO
2826  006204  010237  001142                  MOV     R2,$BDDAT               ;SAVE RESULT FOR TYPE
2827  006210  005037  001140                  CLR     $GDDAT          ;LOAD EXPECTED RESULT
2828  006214  104005                          ERROR   5               ;TRISTATE BUS IS STUCK AT ONE
2829  006216  052702  000001                  BIS     #BIT0,R2        ;SET ERROR FLAG
2830  006222                          10$:
2831
2832  006222  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2833  006230  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2834                        ;PRESET SELECTED REGISTERS TO ZEROS
2835                        ;(ASSUME RMCS1, RMER1, RMER2 WERE CLEARED BY INIT)
2836
2837  006234  012760  000000  000006          MOV     #0,RMDA(R0)     ;LOAD RMDA
2838
2839  006242  012760  000000  000032          MOV     #0,RMOF(R0)     ;LOAD RMOF
2840
2841  006250  012760  000000  000034          MOV     #0,RMDC(R0)     ;LOAD RMDC
2842                        ;WRITE ONES IN SELECTED REGISTERS
2843
2844  006256  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
2845
2846  006264  012760  177777  000006          MOV     #-1,RMDA(R0)    ;LOAD RMDA
2847
2848  006272  012760  016200  000032          MOV     # CXNUOF,RMOF(R0)      ;LOAD RMOF
2849
2850  006300  012760  001777  000034          MOV     # CXNUDC,RMDC(R0)      ;LOAD RMDC
2851
2852  006306  012760  177777  000014          MOV     #-1,RMER1(R0)   ;LOAD RMER1
2853
2854  006314  012760  176210  000042          MOV     # CXNUER2,RMER2(R0)     ;LOAD RMER2
2855                        ;READ ALL REGISTERS
2856
2857  006322  016037  000000  001326          MOV     RMCS1(R0),RMCS1I        ;STORE RMCS1 IN INPUT BUFFER
```

```
2858
2859  006330  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2860
2861  006336  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
2862
2863  006344  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2864
2865  006352  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
2866
2867  006360  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
2868                                  ;CHECK EACH REGISTER CONTENT FOR ONE BITS WRITTEN & READ
2869  006366  042737  177701  001326          BIC     # CILF76,RMCS1I         ;CLEAR ALL BITS NOT WRITTEN
2870  006374  042737  161577  001360          BIC     #XNUOF,RMOFI
2871  006402  042737  176000  001362          BIC     #XNUDC,RMDCI
2872  006410  042737  001567  001370          BIC     #XNUER2,RMER2I
2873  006416  005002                          CLR     R2              ;ACCUMULATE ONES IN R2
2874  006420  053702  001326                  BIS     RMCS1I,R2               ;ACCUMULATE ALL ONE BITS
2875  006424  053702  001334                  BIS     RMDAI,R2
2876  006430  053702  001360                  BIS     RMOFI,R2
2877  006434  053702  001362                  BIS     RMDCI,R2
2878  006440  053702  001342                  BIS     RMER1I,R2
2879  006444  053702  001370                  BIS     RMER2I,R2
2880  006450  022702  177777                  CMP     #-1,R2          ;SEE IF EACH BIT POSITION WAS ONE
2881  006454  001410                          BEQ     20$             ;BRANCH IF NONE STUCK
2882                                  ;ONE OR MORE BIT POSITIONS ARE NOT ONE
2883  006456  010237  001142                  MOV     R2,$BDDAT               ;SAVE RESULT FOR TYPE
2884  006462  012737  177777  001140          MOV     #-1,$GDDAT              ;EXPECTED RESULT
2885  006470  104006                          ERROR   6
2886  006472  052702  000002                  BIS     #BIT1,R2        ;SET ERROR FLAG
2887  006476                          20$:
2888
2889  006476  005702                          TST     R2              ;ANY ERRORS DETECTED ??
2890  006500  001131                          BNE     30$             ;YES - DONT DO BIT TEST
2891  006502  012702  000001                  MOV     #1,R2           ;R2=BIT POSITION
2892  006506                          25$:
2893  006506  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
2894  006514  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
2895                                  ;WRITE THE BIT PATTERN IN SELECTED DEVICE REGISTERS
2896
2897  006520  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
2898
2899  006524  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
2900
2901  006530  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
2902
2903  006534  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
2904
2905  006540  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
2906                                  ;READ BACK THE REGISTERS
2907
2908  006544  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
2909
2910  006552  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
2911
2912  006560  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
2913
```

```
 2914  006566  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
 2915
 2916  006574  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
 2917                                  ;CHECK REGISTER CONTENTS FOR CORRECT PATTERN
 2918  006602  005003                          CLR     R3                      ;R3=ACCUMULATED ONE BIT
 2919  006604  012704  177777                  MOV     #-1,R4                  ;R4=ACCUMULATED ZERO BITS
 2920  006610  013705  001334                  MOV     RMDAI,R5                ;GET ANY GOOD BITS FROM RMDA
 2921  006614  050503                          BIS     R5,R3
 2922  006616  005105                          COM     R5
 2923  006620  040504                          BIC     R5,R4
 2924  006622  013705  001360                  MOV     RMOFI,R5                ;GET GOOD BITS FROM RMOF
 2925  006626  042705  161577                  BIC     #XNUOF,R5
 2926  006632  050503                          BIS     R5,R3
 2927  006634  005105                          COM     R5
 2928  006636  042705  161577                  BIC     #XNUOF,R5
 2929  006642  040504                          BIC     R5,R4
 2930  006644  013705  001362                  MOV     RMDCI,R5                ;GET GOOD BITS FROM RMDC
 2931  006650  042705  176000                  BIC     #XNUDC,R5
 2932  006654  050503                          BIS     R5,R3
 2933  006656  005105                          COM     R5
 2934  006660  042705  176000                  BIC     #XNUDC,R5
 2935  006664  040504                          BIC     R5,R4
 2936  006666  013705  001342                  MOV     RMER1I,R5               ;GET GOOD BITS FROM RMER1
 2937  006672  050503                          BIS     R5,R3
 2938  006674  005105                          COM     R5
 2939  006676  040504                          BIC     R5,R4
 2940  006700  013705  001370                  MOV     RMER2I,R5               ;GET GOOD BITS FROM RMER2
 2941  006704  042705  001567                  BIC     #XNUER2,R5
 2942  006710  050503                          BIS     R5,R3
 2943  006712  005105                          COM     R5
 2944  006714  042705  001567                  BIC     #XNUER2,R5
 2945  006720  040504                          BIC     R5,R4
 2946  006722  010205                          MOV     R2,R5           ;RESET ALL ONES IN R3 EXCEPT
 2947  006724  005105                          COM     R5              ;FOR THE TEST BIT
 2948  006726  040503                          BIC     R5,R3
 2949  006730  040204                          BIC     R2,R4           ;RESET TEST BIT IN R4
 2950  006732  050403                          BIS     R4,R3           ;COMBINE ACCUMULATED 1'S + 0'S
 2951  006734  020302                          CMP     R3,R2           ;IS PATTERN OK??
 2952  006736  001406                          BEQ     26$             ;YES!!
 2953  006740  010237  001140                  MOV     R2,$GDDAT               ;SAVE TEST PATTERN
 2954  006744  010337  001142                  MOV     R3,$BDDAT               ;SAVE RESULT
 2955  006750  104007                          ERROR   7               ;BIT INTERFERENCE IN TRISTATE BUS
 2956  006752  000404                          BR      30$             ;SKIP TO NEXT
 2957  006754                          26$:
 2958                                  ;ADVANCE R2 TO THE NEXT PATTERN AND REPEAT TEST
 2959  006754  006302                          ASL     R2              ;SHIFT THE BIT
 2960  006756  001402                          BEQ     30$             ;EXIT IF DONE
 2961  006760  000137  006506                  JMP     25$
 2962  006764                          30$:
 2963
 2964                                  ;;*******************************************************************
 2965                                  ;*TEST 6         REGISTER SELECT TEST
 2966
 2967                                  ;;*******************************************************************
 2968  006764  000004                  TST6:   SCOPE
 2969  006766  012737  000006  001226          MOV     #6,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
```

```
2970
2971  006774  000240                          NOP
2972  006776  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
2973  007004  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
2974  007012  012737  007026  001122          MOV     #T6,$LPADR  ;LOAD LOOP ON TEST ADDRESS
2975  007020  012737  007026  001124          MOV     #T6,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
2976  007026                          T6:
2977  007026  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
2978  007032  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
2979  007036  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
2980                                  ;THE FOLLOWING TABLE GIVES MASSBUS REGISTER SELECT VALUES FOR
2981                                  ;EACH DEVICE REGISTER
2982
2983                                  ;       REGISTER        REG SEL
2984                                  ;       NAME            (16,8,4,2,1)
2985                                  ;
2986                                  ;       RMCS1           00000
2987                                  ;       RMDS            00001
2988                                  ;       RMER1           00010
2989                                  ;       RMMR1           00011
2990                                  ;       RMAS            00100
2991                                  ;       RMDA            00101
2992                                  ;       RMDT            00110
2993                                  ;       RMLA            00111
2994                                  ;       RMSN            01000
2995                                  ;       RMOF            01001
2996                                  ;       RMDC            01010
2997                                  ;       RMHR            01011
2998                                  ;       MMR2            01100
2999                                  ;       RMER2           01101
3000                                  ;       RMEC1           01110
3001                                  ;       RMEC2           01111
3002
3003                                  ;EACH REGISTER SELECT LINE IS TESTED FOR A STUCK AT ONE,
3004                                  ;STUCK AT ZERO FAULT. AS AN EXAMPLE, TO TEST REG SEL 1.
3005                                  ;FOR S-A-0, RMER1 IS WRITTEN WITH ZEROS. THEN THE REGISTER
3006                                  ;THAT HAS THE SAME SELECT VALUE, EXCEPT FOR THE SELECT LINE
3007                                  ;BEING TESTED, IS WRITTEN WITH ONES. IN THIS EXAMPLE,
3008                                  ;RMMR1 IS WRITTEN WITH ONES. IF SELECT LINE 1 IS S-A-0,
3009                                  ;THE ALL ONES WORD WILL BE WRITTEN IN RMER1, AND RMER1
3010                                  ;WILL NOT BE 0 WHEN READ BACK.
3011
3012  007042  005002                          CLR     R2              ;R2 - ZEROS SOURCE
3013  007044  012703  177777                  MOV     #-1,R3          ;R3 - ONES  SOURCE
3014                                  ;TEST REG SEL 1 FOR S-A-0
3015  007050  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3016  007056  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3017
3018  007062  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3019
3020  007066  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3021
3022  007072  010360  000024                  MOV     R3,RMMR1(R0)    ;LOAD RMMR1
3023
3024  007076  010360  000036                  MOV     R3,RMHR(R0)     ;LOAD RMHR
3025
```

```
3026  007102  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3027
3028  007110  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3029  007116  020337  001342                  CMP     R3,RMER1I
3030  007122  001007                          BNE     10$
3031  007124  052737  176000  001362          BIS     #XNUDC,RMDCI
3032  007132  020337  001362                  CMP     R3,RMDCI
3033  007136  001001                          BNE     10$
3034  007140  104010                          ERROR   10              ;REG SEL 1 IS S-A-0
3035  007142                          10$:
3036
3037                                  ;TEST REG SEL 1 FOR S-A-1
3038  007142  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3039  007150  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3040
3041  007154  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3042
3043  007160  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3044
3045  007164  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3046
3047  007170  010360  000016                  MOV     R3,RMAS(R0)     ;LOAD RMAS
3048
3049  007174  010360  000030                  MOV     R3,RMSN(R0)     ;LOAD RMSN
3050
3051  007200  010360  000040                  MOV     R3,RMMR2(R0)    ;LOAD RMMR2
3052
3053  007204  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3054
3055  007212  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3056
3057  007220  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3058  007226  020337  001334                  CMP     R3,RMDAI
3059  007232  001015                          BNE     20$
3060  007234  052737  161577  001360          BIS     #XNUOF,RMOFI
3061  007242  020337  001360                  CMP     R3,RMOFI
3062  007246  001007                          BNE     20$
3063  007250  052737  001567  001370          BIS     #XNUER2,RMER2I
3064  007256  020337  001370                  CMP     R3,RMER2I
3065  007262  001001                          BNE     20$
3066  007264  104011                          ERROR   11              ;REG SEL 1 IS S-A-1
3067  007266                          20$:
3068
3069                                  ;TEST REG SEL 2 FOR S-A-0
3070  007266  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3071  007274  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3072
3073  007300  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3074
3075  007304  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3076
3077  007310  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3078
3079  007314  010360  000020                  MOV     R3,RMLA(R0)     ;LOAD RMLA
3080
3081  007320  010360  000036                  MOV     R3,RMHR(R0)     ;LOAD RMHR
```

```
3082
3083  007324  010360  000046                  MOV     R3,RMEC2(R0)    ;LOAD RMEC2
3084
3085  007330  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3086
3087  007336  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3088
3089  007344  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3090  007352  020337  001334                  CMP     R3,RMDAI
3091  007356  001015                          BNE     30$
3092  007360  052737  161577  001360          BIS     #XNUOF,RMOFI
3093  007366  020337  001360                  CMP     R3,RMOFI
3094  007372  001007                          BNE     30$
3095  007374  052737  001567  001370          BIS     #XNUER2,RMER2I
3096  007402  020337  001370                  CMP     R3,RMER2I
3097  007406  001001                          BNE     30$
3098  007410  104012                          ERROR   12              ;REG SEL 2 IS S-A-0
3099  007412                          30$:
3100
3101                                  ;TEST REG SEL 2 FOR S-A-1
3102  007412  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3103  007420  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3104
3105  007424  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3106
3107  007430  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3108
3109  007434  012760  000076  000000          MOV     #ILF76,RMCS1(R0)        ;LOAD RMCS1
3110
3111  007442  010360  000030                  MOV     R3,RMSN(R0)     ;LOAD RMSN
3112
3113  007446  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3114
3115  007454  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3116  007462  052737  177701  001342          BIS     # CILF76,RMER1I
3117  007470  020337  001342                  CMP     R3,RMER1I
3118  007474  001007                          BNE     40$
3119  007476  052737  176000  001362          BIS     #XNUDC,RMDCI
3120  007504  020337  001362                  CMP     R3,RMDCI
3121  007510  001001                          BNE     40$
3122  007512  104013                          ERROR   13              ;REG SEL 2 IS S-A-1
3123  007514                          40$:
3124
3125                                  ;TEST REG SEL 4 FOR S-A-0
3126  007514  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3127  007522  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3128
3129  007526  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3130
3131  007532  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3132
3133  007536  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3134
3135  007542  010360  000026                  MOV     R3,RMDT(R0)     ;LOAD RMDT
3136
3137  007546  010360  000042                  MOV     R3,RMER2(R0)    ;LOAD RMER2
```

```
3138
3139  007552  010360  000044                  MOV     R3,RMEC1(R0)    ;LOAD RMEC1
3140
3141  007556  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3142
3143  007564  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3144
3145  007572  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3146  007600  020337  001342                  CMP     R3,RMER1I
3147  007604  001015                          BNE     50$
3148  007606  052737  161577  001360          BIS     #XNUOF,RMOFI
3149  007614  020337  001360                  CMP     R3,RMOFI
3150  007620  001007                          BNE     50$
3151  007622  052737  176000  001362          BIS     #XNUDC,RMDCI
3152  007630  020337  001362                  CMP     R3,RMDCI
3153  007634  001001                          BNE     50$
3154  007636  104014                          ERROR   14              ;REG SEL 4 IS S-A-0
3155  007640                          50$:
3156
3157                                  ;TEST REG SEL 4 FOR S-A-1
3158  007640  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3159  007646  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3160
3161  007652  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3162
3163  007656  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3164
3165  007662  010360  000012                  MOV     R3,RMDS(R0)     ;LOAD RMDS
3166
3167  007666  010360  000032                  MOV     R3,RMOF(R0)     ;LOAD RMOF
3168
3169  007672  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3170
3171  007700  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3172  007706  020337  001334                  CMP     R3,RMDAI
3173  007712  001007                          BNE     60$
3174  007714  052737  001567  001370          BIS     #XNUER2,RMER2I
3175  007722  020337  001370                  CMP     R3,RMER2I
3176  007726  001001                          BNE     60$
3177  007730  104015                          ERROR   15              ;REG SEL 4 IS S-A-1
3178  007732                          60$:
3179
3180                                  ;TEST REG SEL 8 FOR S-A-0
3181  007732  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3182  007740  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3183
3184  007744  010260  000014                  MOV     R2,RMER1(R0)    ;LOAD RMER1
3185
3186  007750  010260  000006                  MOV     R2,RMDA(R0)     ;LOAD RMDA
3187
3188  007754  010360  000034                  MOV     R3,RMDC(R0)     ;LOAD RMDC
3189
3190  007760  010360  000042                  MOV     R3,RMER2(R0)    ;LOAD RMER2
3191
3192  007764  016037  000014  001342          MOV     RMER1(R0),RMER1I        ;STORE RMER1 IN INPUT BUFFER
3193
```

```
3194  007772  016037  000006  001334          MOV     RMDA(R0),RMDAI  ;STORE RMDA IN INPUT BUFFER
3195  010000  020337  001342                  CMP     R3,RMER1I
3196  010004  001004                          BNE     70$
3197  010006  020337  001334                  CMP     R3,RMDAI
3198  010012  001001                          BNE     70$
3199  010014  104016                          ERROR   16              ;REG SEL 8 IS S-A-0
3200  010016                          70$:
3201
3202                                  ;TEST REG SEL 8 FOR S-A-1
3203  010016  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3204  010024  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3205
3206  010030  010260  000032                  MOV     R2,RMOF(R0)     ;LOAD RMOF
3207
3208  010034  010260  000034                  MOV     R2,RMDC(R0)     ;LOAD RMDC
3209
3210  010040  010260  000042                  MOV     R2,RMER2(R0)    ;LOAD RMER2
3211
3212  010044  010360  000012                  MOV     R3,RMDS(R0)     ;LOAD RMDS
3213
3214  010050  010360  000014                  MOV     R3,RMER1(R0)    ;LOAD RMER1
3215
3216  010054  010360  000006                  MOV     R3,RMDA(R0)     ;LOAD RMDA
3217
3218  010060  016037  000032  001360          MOV     RMOF(R0),RMOFI  ;STORE RMOF IN INPUT BUFFER
3219
3220  010066  016037  000034  001362          MOV     RMDC(R0),RMDCI  ;STORE RMDC IN INPUT BUFFER
3221
3222  010074  016037  000042  001370          MOV     RMER2(R0),RMER2I        ;STORE RMER2 IN INPUT BUFFER
3223  010102  052737  161577  001360          BIS     #XNUOF,RMOFI
3224  010110  001015                          BNE     80$
3225  010112  022737  176000  001362          CMP     #XNUDC,RMDCI
3226  010120  020337  001362                  CMP     R3,RMDCI
3227  010124  001007                          BNE     80$
3228  010126  052737  001567  001370          BIS     #XNUER2,RMER2I
3229  010134  020337  001370                  CMP     R3,RMER2I
3230  010140  001001                          BNE     80$
3231  010142  104017                          ERROR   17              ;REG SEL 8 IS S-A-1
3232  010144                          80$:
3233
3234                                  ;REGISTER SELECT 16 IS TESTED BY THE ILR TEST
3235
3236                                  ;;*****************************************************************
3237                                  ;*TEST 7         DRIVE TYPE TEST
3238
3239                                  ;;*****************************************************************
3240  010144  000004                  TST7:   SCOPE
3241  010146  012737  000007  001226          MOV     #7,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
3242
3243  010154  000240                          NOP
3244  010156  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3245  010164  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3246  010172  012737  010206  001122          MOV     #T7,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3247  010200  012737  010206  001124          MOV     #T7,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3248  010206                          T7:
3249  010206  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
```

```
3250  010212  013700  001276            MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
3251  010216  013701  001456            MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
3252
3253  010222  016037  000026  001142    MOV     RMDT(R0),$BDDAT ;STORE RMDT AT $BDDAT
3254  010230  022737  020024  001142    CMP     #SNGPRT,$BDDAT  ;SINGLE PORT RM03??
3255  010236  001432                    BEQ     10$               ;YES!!
3256  010240  022737  024024  001142    CMP     #DULPRT,$BDDAT  ;DUAL PORT RM03??
3257  010246  001426                    BEQ     10$               ;YES!!
3258  010250  022737  020025  001142    CMP     #SNGPRT!BIT0,$BDDAT     ;SINGLE PROT RM02 ?
3259  010256  001422                    BEQ     10$                     ;BRANCH IF SO
3260  010260  022737  024025  001142    CMP     #DULPRT!BIT0,$BDDAT     ;DUAL PORT RM02 ?
3261  010266  001416                    BEQ     10$                     ;BRANCH IF SO
3262  010270  012737  020024  001174    MOV     #SNGPRT,$TMP0   ;LOAD ACCEPTABLE VALUES
3263  010276  012737  024024  001176    MOV     #DULPRT,$TMP1
3264  010304  010037  001136            MOV     R0,$BDADR         ;LOAD BAD ADDRESS
3265  010310  062737  000026  001136    ADD     #RMDT,$BDADR
3266  010316  104057                    ERROR   57                ;DEVICE NOT AN RM03
3267  010320  000137  021416            JMP     $EOSP             ;GO TO NEXT DEVICE
3268  010324                    10$:
3269
3270                            ;;*****************************************************************
3271                            ;*TEST 10        DEVICE AVAILABLE TEST
3272
3273                            ;;*****************************************************************
3274  010324  000004            TST10:  SCOPE
3275  010326  012737  000010  001226    MOV     #10,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
3276
3277  010334  000240                    NOP
3278  010336  012737  000024  001120    MOV     #20.,$ICNT        ;20 ITERATIONS
3279  010344  112737  000001  001131    MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
3280  010352  012737  010366  001122    MOV     #T10,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3281  010360  012737  010366  001124    MOV     #T10,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3282  010366                    T10:
3283  010366  012706  001100            MOV     #STACK,SP         ;LOAD THE STACK POINTER
3284  010372  013700  001276            MOV     $BASE,R0          ;R0 = UNIBUS ADDRESS OF UUT
3285  010376  013701  001456            MOV     TSTQUE,R1         ;R1 = POINTER TO DEVICE
3286  010402  012760  000040  000010    MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3287  010410  111160  000010            MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3288
3289  010414  016037  000000  001142    MOV     RMCS1(R0),$BDDAT        ;STORE RMCS1 AT $BDDAT
3290  010422  042737  173777  001142    BIC     # CDVA,$BDDAT   ;CLEAR ALL BUT DVA
3291  010430  001006                    BNE     10$               ;BRANCH IF DVA SET
3292  010432  012737  004000  001140    MOV     #DVA,$GDDAT     ;SETUP EXPECTED
3293  010440  010037  001136            MOV     R0,$BDADR         ;SETUP REG ADDRESS
3294  010444  104060                    ERROR   60                ;DEVICE NOT AVAILABLE
3295  010446                    10$:
3296
3297                            ;;*****************************************************************
3298                            ;*TEST 11        SEARCH TIMEOUT TEST
3299
3300                            ;;*****************************************************************
3301  010446  000004            TST11:  SCOPE
3302  010450  012737  000011  001226    MOV     #11,$TESTN        ;;SET TEST NUMBER IN APT MAIL BOX
3303  010456  000240                    NOP
3304  010460  012737  000024  001120    MOV     #20.,$ICNT        ;20 ITERATIONS
3305  010466  112737  000001  001131    MOVB    #1,$ERMAX         ;ONE ERROR ALLOWED
```

```
3306  010474  012737  010510  001122          MOV     #T11,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3307  010502  012737  010510  001124          MOV     #T11,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3308  010510                              T11:
3309  010510  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3310  010514  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3311  010520  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3312                                    ;LOAD REGISTER OUTPUT BUFFER WITH COMMAND PARAMETERS
3313  010524  012737  000000  001410          MOV     #0,RMDA0        ;SECTOR=0=TRACK
3314  010532  012737  000000  001436          MOV     #0,RMDC0        ;CYLINDER=0
3315  010540  012737  010000  001434          MOV     #FMT16,RMOF0    ;16 BIT FORMAT
3316  010546  012737  054062  001406          MOV     #BUFFER,RMBA0   ;STARTING BUFFER ADDRESS
3317  010554  012737  177777  001404          MOV     # C1+1,RMWC0   ;WORD COUNT
3318  010562  012737  000061  001402          MOV     #WD!GO,RMCS10   ;WRITE DATA COMMAND
3319
3320                                    ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3321  010570  004737  023326                  JSR     PC,ENBSCH
3322  010574  000402                          BR      10$             ;GO TO 10$ IF NO ERROR
3323  010576  104000                          ERROR                   ;RETURN HERE IF ERROR
3324  010600  000462                          BR      50$             ;SLO  REST PF TEST
3325  010602                              10$:
3326                                    ;START THE CLOCK AND WAIT FOR 100 MS
3327  010602  012737  000144  001524          MOV     #100.,WATCH     ;SET WATCHDOG TIMER VALUE
3328  010610  004777  170712                  JSR     PC,@CLOCK       ;START THE CLOCK
3329  010614  005737  001524              20$:TST     WATCH
3330  010620  001375                          BNE     20$
3331  010622  004777  170702                  JSR     PC,@STOP        ;STOP THE CLOCK
3332                                    ;VERIFY THAT OPI IS NOT SET (SEARCH TIMEOUT IS DISABLED)
3333
3334  010626  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3335  010634  010037  001136                  MOV     R0,$BDADR       ;SET UP FOR ERROR MSG
3336  010640  062737  000014  001136          ADD     #RMER1,$BDADR
3337  010646  042737  157777  001142          BIC     # COPI,$BDDAT
3338  010654  001404                          BEQ     30$             ;BRANCH IF NO ERROR
3339  010656  005037  001140                  CLR     $GDDAT
3340  010662  104020                          ERROR   20                      ;OPI SET W/SEARCH TIMEOUT
3341                                                                  ;DISABLED
3342  010664  000430                          BR      50$             ;SKIP
3343  010666                              30$:
3344                                    ;ENABLE SEARCH TIMEOUT, THEN WAIT 100 MS
3345
3346  010666  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
3347  010674  012737  000144  001524          MOV     #100.,WATCH     ;SET WATCHDOG TIMER VALUE
3348  010702  004777  170620                  JSR     PC,@CLOCK       ;START THE CLOCK
3349  010706  005737  001524              40$:TST     WATCH
3350  010712  001375                          BNE     40$
3351  010714  004777  170610                  JSR     PC,@STOP        ;STOP THE CLOCK
3352                                    ;OPI SHOULD NOW BE SET (SEARCH TIMEOUT IS ENABLED)
3353
3354  010720  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3355  010726  042737  157777  001142          BIC     # COPI,$BDDAT
3356  010734  001004                          BNE     50$
3357  010736  012737  020000  001140          MOV     #OPI,$GDDAT
3358  010744  104021                          ERROR   21                      ;OPI NOT SET BY SEARCH TIMEOUT
3359  010746                              50$:
3360
3361                                    ;;*****************************************************************
```

```
                                        ;*TEST 12       SET DTE TEST
                                        ;;*****************************************************************
                                        TST12:  SCOPE
                                                MOV     #12,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
                                                NOP
                                                MOV     #20.,$ICNT      ;20 ITERATIONS
                                                MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
                                                MOV     #T12,$LPADR  ;LOAD LOOP ON TEST ADDRESS
                                                MOV     #T12,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
                                        T12:
                                                MOV     #STACK,SP       ;LOAD THE STACK POINTER
                                                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
                                                MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
                                                MOV     R0,$BDADR       ;SETUP ERROR MSG
                                                ADD     #RMER1,$BDADR
                                                CLR     $GDDAT
                                        ;INITILAIZE AND VERIFY THAT DRIVE TIMING ERROR IS RESET
                                                MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
                                                MOVB    (R1),RMCS2(R0)  ;SELECT UNIT

                                                MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
                                                BIC     # CDTE,$BDDAT
                                                BEQ     10$             ;BRANCH IF DTE=0
                                                ERROR   31                      ;CANT RESET DTE
                                                BR      50$             ;SKIP REST OF TEST
                                        ;SET MAINTENANCE INDEX PULSE AND VERIFY DTE REMAINS RESET
                                        10$:

                                                MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1

                                                MOV     #DMD!MI,RMMR1(R0)       ;LOAD RMMR1

                                                MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
                                                BIC     # CDTE,$BDDAT
                                                BEQ     20$
                                                ERROR                   ;DTE SET WHEN SECTOR
                                                BR      50$             ;COMPARE SHOULD BE RESET
                                        20$:
                                        ;EXECUTE DUMMY DATA COMMAND TO ENABLE SEARCH
                                                MOV     #0,RMDA0
                                                MOV     #5,RMDC0
                                                MOV     #FMT16,RMOF0
                                                MOV     #BUFFER,RMBA0
                                                MOV     # C1+1,RMWC0
                                                MOV     #WD!GO,RMCS10

                                        ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
                                                JSR     PC,ENBSCH
                                                BR      30$             ;GO TO 30$ IF NO ERROR
                                                ERROR                   ;RETURN HERE IF ERROR
                                                BR      50$
                                        ;WITH SEARCH ENABLED, SET AND RESET SECTOR PULSE TO SET ENABLE
                                        ;SEARCH FLOP.
                                        30$:
```

```
 3418  011212  012760  051441  000024          MOV     #DMD:MUR!DBEN!MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
 3419
 3420  011220  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
 3421                                  ;SET SECTOR PULSE AND VERIFY DTE DOES NOT SET
 3422                                  ;       PUTMR1  #DMD!MUR!DBEN!MOC!DTO!MS
 3423                                  ;       PUTMR1  #DMD!MUR!DBEN!MOC!DTO
 3424
 3425  011226  016037  000014  001142          MOV     RMER1(R0),$BDDAT         ;STORE RMER1 AT $BDDAT
 3426  011234  042737  167777  001142          BIC     # CDTE,$BDDAT
 3427  011242  001402                          BEQ     40$
 3428  011244  104023                          ERROR   23               ;DTE SET WHEN SECTOR
 3429  011246  001432                          BEQ     50$              ;COMPARE SHOULD BE RESET
 3430  011250                          40$:
 3431                                  ;FORCE SECTOR COMPARE
 3432
 3433  011250  012760  051403  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)      ;LOAD RMMR1
 3434
 3435  011256  012760  051443  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
 3436
 3437  011264  012760  051403  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)      ;LOAD RMMR1
 3438                                  ;SET SECTOR PULSE AND VERIFY DTE SETS
 3439
 3440  011272  012760  051441  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
 3441
 3442  011300  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
 3443
 3444  011306  016037  000014  001142          MOV     RMER1(R0),$BDDAT         ;STORE RMER1 AT $BDDAT
 3445  011314  042737  167777  001142          BIC     # CDTE,$BDDAT
 3446  011322  001004                          BNE     50$
 3447  011324  012737  010000  001142          MOV     #DTE,$BDDAT
 3448  011332  104024                          ERROR   24                       ;COULD NOT SET DTE WITH
 3449                                                                   ;SECTOR COMPARE SET
 3450  011334                          50$:
 3451
 3452                                  ;;*****************************************************************
 3453                                  ;*TEST 13        FORMAT CHANGE TEST
 3454
 3455                                  ;;*****************************************************************
 3456  011334  000004                  T$T13:  SCOPE
 3457  011336  012737  000013  001226          MOV     #13,$TESTN       ;;SET TEST NUMBER IN APT MAIL BOX
 3458  011344  000240                          NOP
 3459  011346  012737  000024  001120          MOV     #20.,$ICNT       ;20 ITERATIONS
 3460  011354  112737  000001  001131          MOVB    #1,$ERMAX        ;ONE ERROR ALLOWED
 3461  011362  012737  011376  001122          MOV     #T13,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 3462  011370  012737  011376  001124          MOV     #T13,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 3463  011376                          T13:
 3464  011376  012706  001100                  MOV     #STACK,SP        ;LOAD THE STACK POINTER
 3465  011402  013700  001276                  MOV     $BASE,R0         ;R0 = UNIBUS ADDRESS OF UUT
 3466  011406  013701  001456                  MOV     TSTQUE,R1        ;R1 = POINTER TO DEVICE
 3467  011412  012702  011752                  MOV     #50$,R2          ;R2=TABLE POINTER
 3468  011416                          10$:
 3469                                  ;;*****************************************************************
 3470                                  ;INITILAIZE AND SET THE FORMAT BIT, USE INDEX PULSE TO CLEAR FORMAT CHANGE
 3471  011416  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 3472  011424  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
 3473
```

```
3474  011430  011260  000032                MOV     (R2),RMOF(R0)    ;LOAD RMOF
3475
3476  011434  012760  000001  000024        MOV     #DMD,RMMR1(R0)   ;LOAD RMMR1
3477
3478  011442  012760  000005  000024        MOV     #DMD!MI,RMMR1(R0)       ;LOAD RMMR1
3479                                ;SETUP AND EXECUTE DUMMY DATA COMMAND USING OPPOSITE FORMAT
3480  011450  012737  000000  001410        MOV     #0,RMDAO
3481  011456  012737  000005  001436        MOV     #5,RMDCO
3482  011464  016237  000002  001434        MOV     2(R2),RMOFO
3483  011472  012737  054062  001406        MOV     #BUFFER,RMBAO
3484  011500  012737  177777  001404        MOV     # C1+1,RMWCO
3485  011506  012737  000061  001402        MOV     #WD!GO,RMCS10
3486
3487                                ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3488  011514  004737  023326                JSR     PC,ENBSCH
3489  011520  000402                        BR      20$             ;GO TO 20$ IF NO ERROR
3490  011522  104000                        ERROR                   ;RETURN HERE IF ERROR
3491  011524  000515                        BR      60$
3492  011526                        20$:
3493                                ;FORMAT CHANGE FLOP SHOULD BE SET - VERIFY BY TRYING TO FORCE A
3494                                ;DRIVE TIMING ERROR WHICH SHOULD NOT SET.
3495
3496  011526  012760  051403  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3497
3498  011534  012760  051443  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
3499
3500  011542  012760  051403  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3501
3502  011550  012760  051401  000024        MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3503                                ;VERIFY THAT DRIVE TIMING ERROR DIDNT SET
3504
3505  011556  016037  000014  001142        MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3506  011564  042737  167777  001142        BIC     # CDTE,$BDDAT
3507  011572  001416                        BEQ     30$
3508  011574  010037  001136                MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3509  011600  062737  000014  001136        ADD     #RMER1,$BDADR
3510  011606  005037  001140                CLR     $GDDAT
3511  011612  011237  001174                MOV     (R2),$TMP0
3512  011616  016237  000002  001176        MOV     2(R2),$TMP1
3513  011624  104025                        ERROR   25                      ;DTE SET WHEN THERE WAS
3514  011626  000454                        BR      60$             ;A FORMAT CHANGE
3515  011630                        30$:
3516                                ;CLEAR THE FORMAT CHANGE FLOP W/INDEX PULSE
3517
3518  011630  012760  051405  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MI,RMMR1(R0)       ;LOAD RMMR1
3519                                ;ENABLE SEARCH AND FORCE SECTOR COMPARE
3520
3521  011636  012760  051403  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3522
3523  011644  012760  051443  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
3524
3525  011652  012760  051403  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
3526                                ;SET DTE W/ANOTHER SECTOR PULSE - VERIFY DTE IS SET
3527
3528  011660  012760  051441  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
3529
```

```
3530  011666  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3531
3532  011674  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
3533  011702  042737  167777  001142          BIC     # CDTE,$BDDAT
3534  011710  001012                          BNE     40$
3535  011712  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3536  011716  062737  000014  001136          ADD     #RMER1,$BDADR
3537  011724  012737  010000  001140          MOV     #DTE,$GDDAT
3538  011732  104027                          ERROR   27                      ;COULD NOT SET DTE
3539  011734  000411                          BR      60$
3540  011736                          40$:
3541                                  ;;**********************************************************************
3542                                  ;DO TEST W/18 TO 16 FORMAT CHANGE DURING SECOND EXECUTION
3543  011736  022702  011752                  CMP     #50$,R2
3544  011742  001006                          BNE     60$             ;BRANCH IF DONE TEST
3545  011744  062702  000002                  ADD     #2,R2           ;MOVE POINTER
3546  011750  000622                          BR      10$             ;REPEAT TEST
3547
3548  011752  010000                  50$:    .WORD   #FMT16          ;16-18 FORMAT CHANGE
3549  011754  000000                          .WORD   0               ;18-16 FORMAT CHANGE
3550  011756  010000                          .WORD   #FMT16
3551  011760                          60$:                            ;END OF TEST
3552
3553                                  ;;**********************************************************************
3554                                  ;*TEST 14        PROM STROBE TEST
3555
3556                                  ;;**********************************************************************
3557  011760  000004                  TST14:  SCOPE
3558  011762  012737  000014  001226          MOV     #14,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3559  011770  000240                          NOP
3560  011772  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3561  012000  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3562  012006  012737  012022  001122          MOV     #T14,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3563  012014  012737  012022  001124          MOV     #T14,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3564  012022                          T14:
3565  012022  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3566  012026  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3567  012032  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3568  012036  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3569  012044  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3570  012050  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3571  012054  062737  000024  001136          ADD     #RMMR1,$BDADR
3572                                  ;SET DIAGNOSTIC MODE AND TRY TO RESET PROM STROBE
3573
3574  012062  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3575  012070  012702  000021                  MOV     #17.,R2         ;R2=MAXIMUM # CLOCK PULSES
3576  012074                          5$:
3577
3578  012074  012760  004001  000024          MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3579
3580  012102  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3581
3582  012110  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3583  012116  042737  177737  001142          BIC     # CWC,$BDDAT
3584  012124  001406                          BEQ     6$
3585  012126  005302                          DEC     R2
```

```
3586  012130  001361                        BNE     5$
3587  012132  005037  001140                CLR     $GDDAT
3588  012136  104000                        ERROR                   ;CAN'T CLEAR PROM STROBE
3589  012140  000501                        BR      60$
3590  012142  012702  000021        6$:     MOV     #17.,R2
3591  012146                        10$:
3592
3593  012146  012760  004001  000024        MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3594
3595  012154  012760  000001  000024        MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3596
3597  012162  016037  000024  001142        MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3598  012170  042737  177737  001142        BIC     # CWC,$BDDAT
3599  012176  001007                        BNE     20$             ;EXIT LOOP WHEN PROM STROBE ON
3600  012200  005302                        DEC     R2
3601  012202  001361                        BNE     10$
3602  012204  012737  000040  001140        MOV     #WC,$GDDAT
3603  012212  104030                        ERROR   30                      ;CANT SET PROM STROBE
3604  012214  000453                        BR      60$
3605  012216                        20$:
3606                                ;VERIFY PROM STROBE IS ON FOR 3 BIT CLOCKS
3607  012216  012702  000003                MOV     #3.,R2
3608  012222                        25$:
3609
3610  012222  012760  004001  000024        MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3611
3612  012230  012760  000001  000024        MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3613
3614  012236  016037  000024  001142        MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3615  012244  042737  177737  001142        BIC     # CWC,$BDDAT
3616  012252  001005                        BNE     30$             ;BRANCH IF PROM STORBE DID NOT
3617                                                                ;DROP EARLY
3618  012254  012737  000040  001140        MOV     #WC,$GDDAT
3619  012262  104032                        ERROR   32
3620  012264  000427                        BR      60$
3621  012266  005302                30$:    DEC     R2
3622  012270  001354                        BNE     25$
3623                                ;VERIFY PROM STROBE IS OFF FOR 8 BIT CLOCKS
3624  012272  012702  000010                MOV     #8.,R2
3625  012276                        40$:
3626
3627  012276  012760  004001  000024        MOV     #DMD!MCLK,RMMR1(R0)     ;LOAD RMMR1
3628
3629  012304  012760  000001  000024        MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
3630
3631  012312  016037  000024  001142        MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3632  012320  042737  177737  001142        BIC     # CWC,$BDDAT
3633  012326  001404                        BEQ     50$             ;BRANCH IF PROM STROBE
3634  012330  005037  001140                CLR     $GDDAT          ;DID NOT SET EARLY
3635  012334  104033                        ERROR   33
3636  012336  000402                        BR      60$
3637  012340  005302                50$:    DEC     R2
3638  012342  001355                        BNE     40$
3639  012344                        60$:
3640
3641                                ;;*****************************************************************
```

```
3642                                        ;*TEST 15       SYNC WORD COUNT INHIBIT TEST
3643
3644                                        ;;*****************************************************************
3645  012344  000004                        TST15:  SCOPE
3646  012346  012737  000015  001226                MOV     #15,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3647  012354  000240                                NOP
3648  012356  012737  000024  001120                MOV     #20.,$ICNT      ;20 ITERATIONS
3649  012364  112737  000001  001131                MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3650  012372  012737  012406  001122                MOV     #T15,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3651  012400  012737  012406  001124                MOV     #T15,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3652  012406                                T15:
3653  012406  012706  001100                        MOV     #STACK,SP       ;LOAD THE STACK POINTER
3654  012412  013700  001276                        MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3655  012416  013701  001456                        MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3656                                        ;SETUP kEGISTER OUTPUT BUFFER FOR SUBROUTINES
3657  012422  012737  000000  001410                MOV     #0,RMDA0        ;SECTOR=0, TRACK=0
3658  012430  012737  000005  001436                MOV     #5,RMDC0        ;CYLINDER=0
3659  012436  012737  010000  001434                MOV     #FMT16,RMOF0    ;16 BIT FORMAT
3660  012444  012737  054062  001406                MOV     #BUFFER,RMBA0   ;STARTING BUFFER ADDRESS
3661  012452  012737  177777  001404                MOV     # C1+1,RMWC0    ;WORD COUNT
3662  012460  012737  000061  001402                MOV     #WD!GO,RMCS10   ;WRITE DATA COMMAND
3663  012466  012760  000040  000010                MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
3664  012474  111160  000010                       MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
3665
3666                                        ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3667  012500  004737  023326                        JSR     PC,ENBSCH
3668  012504  000402                                BR      10$             ;GO TO 10$ IF NO ERROR
3669  012506  104000                                ERROR                   ;RETURN HERE IF ERROR
3670  012510  000562                                BR      120$            ;SKIP IF FAILURE
3671  012512                                10$:
3672                                        ;FORCE SECTOR COMPARE USING SUBROUTINE
3673  012512  004737  024206                        JSR     PC,SCTCMP
3674  012516  000402                                BR      20$             ;GO TO 20$ IF NO ERROR
3675  012520  104000                                ERROR                   ;RETURN HERE IF ERROR
3676  012522  000555                                BR      120$
3677                                        ;;*****************************************************************
3678                                        ;CAN NOW STEP DATA TIMING SEQUENCER WITH SECTOR COMPARE SET
3679
3680                                        ;STEP THE SEQUENCER TO LOCATION 10(10) WHILE CHECKING "PLFS"
3681  012524  012702  000000                20$:    MOV     #0,R2           ;R2=SEQUENCER ADDRESS
3682  012530                                30$:
3683                                        ;PULSE MAINTENANCE CLOCK UNTIL PROM STROBE SETS
3684
3685  012530  012760  055401  000024                MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3686
3687  012536  012760  051401  000024                MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3688                                        ;SEE IF PROM STROBE IS SET
3689
3690  012544  016003  000024                        MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
3691  012550  032703  000040                        BIT     #WC,R3
3692  012554  001765                                BEQ     30$             ;ISSUE NEXT BIT CLOCK
3693  012556  005202                                INC     R2              ;INCREMENT SEQUENCER ADDRESS
3694                                        ;CHECK "LOOKING FOR SYNC" - SHOULD BE ZERO UNTIL LOCATION 11
3695  012560  010337  001142                        MOV     R3,$BDDAT
3696  012564  042737  175777  001142                BIC     # CPLFS,$BDDAT
3697  012572  001413                                BEQ     50$             ;BRANCH IF PLFS IS ZERO
```

```
3698  012574  005037  001140                  CLR     $GDDAT
3699  012600  010037  001136                  MOV     R0,$BDADR
3700  012604  062737  000024  001136          ADD     #RMMR1,$BDADR
3701  012612  010237  001174                  MOV     R2,$TMP0
3702  012616  104034                          ERROR   34                      ;PLFS SHOULD NOT BE SET
3703  012620  000516                          BR      120$            ;SKIP REST OF TEST
3704  012622                          50$:
3705                                  ;EXIT LOOP IF SEQUENCER NOW AT LOCATION 10
3706  012622  022702  000012                  CMP     #10.,R2
3707  012626  001414                          BEQ     70$
3708                                  ;PULSE MAINTENANCE CLOCK UNTIL PROM STROBE RESETS
3709  012630                          60$:
3710
3711  012630  012760  055401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3712
3713  012636  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3714                                  ;SEE IF PROM STROBE IS RESET
3715
3716  012644  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
3717  012650  032703  000040                  BIT     #WC,R3
3718  012654  001365                          BNE     60$
3719  012656  000724                          BR      30$             ;GO STEP SEQUENCER TO NEXT LOC
3720  012660                          70$:
3721                                  ;;*****************************************************************
3722                                  ;THE NEXT PROM STROBE SHOULD SET "PLFS" AT LOCATION 11(10) OF THE
3723                                  ;DATA TIMING SEQUENCER.
3724  012660  012702  000020                  MOV     #16.,R2         ;R2=NUMBER OF BIT CLOCKS
3725                                  ;ISSUE 16 BIT CLOCKS TO GET NEXT PROM STROBE
3726  012664                          80$:
3727
3728  012664  012760  055401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3729
3730  012672  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3731  012700  005302                          DEC     R2
3732  012702  001370                          BNE     80$
3733                                  ;VERIFY THAT "PLFS" IS NOW SET
3734
3735  012704  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3736  012712  042737  175777  001142          BIC     # CPLFS,$BDDAT
3737  012720  001012                          BNE     90$             ;BRANCH IF PLFS IS SET
3738  012722  012737  002000  001140          MOV     #PLFS,$GDDAT    ;SETUP ERROR MESSAGE
3739  012730  010037  001136                  MOV     R0,$BDADR
3740  012734  062737  000024  001136          ADD     #RMMR1,$BDADR
3741  012742  104035                          ERROR   35                      ;"PLFS" DID NOT SET
3742  012744  000444                          BR      120$            ;SKIP REST OF TEST
3743  012746                          90$:
3744
3745                                  ;ISSUE 3 MORE BIT CLOCKS TO RESET PROM STROBE
3746
3747  012746  012702  000003                  MOV     #3,R2           ;
3748  012752                          95$:
3749
3750  012752  012760  055401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3751
3752  012760  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3753  012766  005302                          DEC     R2
```

```
3754  012770  001370                        BNE     95$
3755                                ;;*****************************************************************
3756                                ;WITH "LOOKING FOR SYNC SET, FURTHER BIT CLOCKS SHOULD NOT SET
3757                                ;PROM STROBE
3758  012772  012702  000040                MOV     #32.,R2         ;R2=NUMBER OF BIT CLOCKS
3759  012776                        100$:
3760                                ;PULSE BIT CLOCK AND VERIFY PROM STROBE DOES NOT SET
3761
3762  012776  012760  055401  000024        MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
3763
3764  013004  012760  051401  000024        MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
3765
3766  013012  016003  000024                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
3767  013016  042703  177737                BIC     # CWC,R3
3768  013022  001413                        BEQ     110$            ;BRANCH IF PROM STROBE IS 0
3769  013024  005037  001140                CLR     $GDDAT          ;SETUP ERROR MESSAGE
3770  013030  010337  001142                MOV     R3,$BDDAT
3771  013034  010037  001136                MOV     R0,$BDADR
3772  013040  062737  000024  001136        ADD     #RMMR1,$BDADR
3773  013046  104036                        ERROR   36                      ;PROM STROBE WHILE LOOKING FOR SYNC
3774  013050  000402                        BR      120$
3775  013052  005302                110$:   DEC     R2              ;CONTINUE FOR 32 BIT CLOCKS
3776  013054  001350                        BNE     100$
3777
3778  013056                        120$:                           ;END OF TEST
3779                                ;;*****************************************************************
3780                                ;*TEST 16        SYNC DETECTION TEST
3781
3782                                ;;*****************************************************************
3783  013056  000004                TST16:  SCOPE
3784  013060  012737  000016  001226        MOV     #16,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3785  013066  000240                        NOP
3786  013070  012737  000024  001120        MOV     #20.,$ICNT      ;20 ITERATIONS
3787  013076  112737  000001  001131        MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3788  013104  012737  013120  001122        MOV     #T16,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3789  013112  012737  013120  001124        MOV     #T16,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3790  013120                        T16:
3791  013120  012706  001100                MOV     #STACK,SP       ;LOAD THE STACK POINTER
3792  013124  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3793  013130  013701  001456                MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3794                                ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
3795  013134  012737  000000  001410        MOV     #0,RMDA0        ;SECTOR 0,TRACK 0
3796  013142  012737  000005  001436        MOV     #5,RMDC0        ;CYLINDER 5
3797  013150  012737  010000  001434        MOV     #FMT16,RMOF0    ;16 BIT MODE
3798  013156  012737  054062  001406        MOV     #BUFFER,RMBA0   ;BUFFER ADDRESS
3799  013164  012737  177777  001404        MOV     # C1+1,RMWC0    ;CORD COUNT
3800  013172  012737  000071  001402        MOV     #RD!GO,RMCS10   ;READ DATA COMMAND
3801
3802                                ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3803  013200  004737  023326                JSR     PC,ENBSCH
3804  013204  000402                        BR      10$             ;GO TO 10$ IF NO ERROR
3805  013206  104000                        ERROR                   ;RETURN HERE IF ERROR
3806  013210  000542                        BR      100$
3807  013212                        10$:
3808                                ;FORCE SECTOR COMPARE USING SUBROUTINE
3809  013212  004737  024206                JSR     PC,SCTCMP
```

```
3810  013216  000402                          BR      20$             ;GO TO 20$ IF NO ERROR
3811  013220  104000                          ERROR                   ;RETURN HERE IF ERROR
3812  013222  000535                          BR      100$
3813  013224                          20$:
3814                                  ;SET "LOOKING FOR SYNC" USING SUBROUTINE
3815  013224  004737  024314                  JSR     PC,SETLFS
3816  013230  000402                          BR      30$
3817  013232  104000                          ERROR
3818  013234  000530                          BR      100$
3819  013236                          30$:
3820                                  ;;*****************************************************************
3821                                  ;CLOCK ALL ONES SYNC PATTERN AND VERIFY SYNC IS NOT DETECTED
3822                                  ;(USING PROM STORBE AS INDICATION)
3823
3824  013236  012702  000020                  MOV     #16.,R2         ;NUMBER OF ONE BITS
3825  013242  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
3826  013246  062737  000024  001136          ADD     #RMMR1,$BDADR
3827  013254  005037  001140                  CLR     $GDDAT
3828
3829  013260  012760  053401  000024          MOV     #MR1AAA!MRD,RMMR1(R0)   ;LOAD RMMR1
3830  013266                          40$:
3831
3832  013266  012760  057401  000024          MOV     #MR1AAA!MRD!MCLK,RMMR1(R0)      ;LOAD RMMR1
3833
3834  013274  012760  053401  000024          MOV     #MR1AAA!MRD,RMMR1(R0)   ;LOAD RMMR1
3835
3836  013302  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3837  013310  042737  177737  001142          BIC     # CWC,$BDDAT
3838  013316  001405                          BEQ     50$
3839  013320  012737  177777  001174          MOV     #177777,$TMP0
3840  013326  104037                          ERROR   37
3841  013330  000472                          BR      100$
3842  013332                          50$:
3843  013332  005302                          DEC     R2              ;REPEAT FOR 16 BITS
3844  013334  001354                          BNE     40$
3845  013336  012702  000020                  MOV     #16.,R2         ;NUMBER OF ZERO BITS
3846
3847                                  ;;*****************************************************************
3848                                  ;CLOCK ALL ZERO SYNC PATTERN AND VERIFY SYNC IS NOT DETECTED
3849  013342                          60$:
3850
3851  013342  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
3852
3853  013350  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
3854
3855  013356  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3856  013364  042737  177737  001142          BIC     # CWC,$BDDAT    ;MAKE SURE SYNC NOT DETECTED
3857  013372  001404                          BEQ     70$
3858  013374  005037  001174                  CLR     $TMP0
3859  013400  104037                          ERROR   37                      ;SYNC DETECTED W/ALL 0'S PATTERN
3860  013402  000445                          BR      100$            ;SKIP IF FAILURE
3861  013404                          70$:
3862  013404  005302                          DEC     R2              ;REPEAT FOR 16 BITS
3863  013406  001355                          BNE     60$
3864  013410  012702  000040                  MOV     #32.,R2         ;R2=NUMBER OF BITS
3865  013414  012703  000031                  MOV     #000031,R3      ;R3=SYNC PATTERN FOR LEFT SHIFT
```

```
3866
3867                                    ;;*****************************************************************
3868                                    ;CLOCK THE BINARY SYNC PATTERN (10011000) AND VERIFY THAT SYNC IS
3869                                    ;DETECTED
3870  013420  012737  055401  001426    80$:    MOV     #MR1AAA!MCLK,RMMR10     ;GENERATE VALUE OF RMMR1
3871  013426  000241                            CLC                     ;CLEAR THE CARR
3872  013430  006003                            ROR     R3              ;SHIFT RIGHT
3873  013432  103003                            BCC     90$             ;BRANCH IF C BIT CLEAR
3874  013434  052737  002000  001426            BIS     #MRD,RMMR10     ;SET MRD IF PATTER BIT SETS
3875  013442                            90$:
3876
3877  013442  013760  001426  000024            MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
3878  013450  042737  004000  001426            BIC     #MCLK,RMMR10            ;RESET BIT CLOCK
3879
3880  013456  013760  001426  000024            MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
3881
3882  013464  016037  000024  001142            MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3883  013472  042737  177737  001142            BIC     # CWC,$BDDAT
3884  013500  001006                            BNE     100$            ;BRANCH IF PROM STROBE IS SET
3885  013502  005302                            DEC     R2              ;CONTINME FOR 16 BIT CLOCKS
3886  013504  001345                            BNE     80$
3887  013506  012737  000040  001140            MOV     #WC,$GDDAT
3888  013514  104040                            ERROR   40                      ;SYNC PATTERN NOT DETECTED
3889  013516                            100$:                           ;END OF TEST
3890
3891                                    ;;*****************************************************************
3892                                    ;*TEST 17       ABORT SYNC DETECTION TEST
3893
3894                                    ;;*****************************************************************
3895  013516  000004                    TST17:  SCOPE
3896  013520  012737  000017  001226            MOV     #17,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3897  013526  000240                            NOP
3898  013530  012737  000024  001120            MOV     #20.,$ICNT      ;20 ITERATIONS
3899  013536  112737  000001  001131            MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3900  013544  012737  013560  001122            MOV     #T17,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3901  013552  012737  013560  001124            MOV     #T17,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3902  013560                            T17:
3903  013560  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
3904  013564  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3905  013570  013701  001456                    MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3906                                    ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
3907  013574  012737  000000  001410            MOV     #0,RMDA0        ;SECTOR 0, TRACK 0
3908  013602  012737  000005  001436            MOV     #5,RMDC0        ;CYLINDER 5
3909  013610  012737  010000  001434            MOV     #FMT16,RMOF0    ;16 BIT FORMAT
3910  013616  012737  054062  001406            MOV     #BUFFER,RMBA0   ;STARTING BUFFER ADDRESS
3911  013624  012737  177777  001404            MOV     # C1+1,RMWC0   ;WORD COUNT
3912  013632  012737  000071  001402            MOV     #RD!GO,RMCS10   ;READ DATA COMMAND
3913
3914                                    ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
3915  013640  004737  023326                    JSR     PC,ENBSCH
3916  013644  000402                            BR      10$             ;GO TO 10$ IF NO ERROR
3917  013646  104000                            ERROR                   ;RETURN HERE IF ERROR
3918  013650  000447                            BR      50$
3919  013652                            10$:
3920                                    ;FORCE SECTOR COMPARE USING SUBROUTINE
3921  013652  004737  024206                    JSR     PC,SCTCMP
```

```
3922  013656  000402                          BR      20$             ;GO TO 20$ IF NO ERROR
3923  013660  104000                          ERROR                   ;RETURN HERE IF ERROR
3924  013662  000442                          BR      50$
3925  013664                          20$:
3926                                  ;SET "LOOKING FOR SYNC" USING SUBROUTINE
3927  013664  004737  024314                  JSR     PC,SETLFS
3928  013670  000402                          BR      30$
3929  013672  104000                          ERROR
3930  013674  000435                          BR      50$
3931  013676                          30$:
3932                                  ;;*****************************************************************
3933                                  ;LOOKING FOR SYNC INHIBITS WORD CLOCK,HOWEVER, "DRIVE TIMING ERROR"
3934                                  ;SHOULD RESET "LFS WORD COUNT INHIBIT" FLOP
3935                                  ;FORCE DRIVE TIMING ERROR
3936
3937
3938  013676  012760  051441  000024          MOV     #MR1AAA!MS,RMMR1(R0)    ;LOAD RMMR1
3939                                  ;PULSE BIT CLOCK AND VERIFY PROM STROBE SETS
3940  013704  012702  000020                  MOV     #16.,R2         ;R2, NUMBER OF BIT CLOCKS
3941  013710                          40$:
3942
3943  013710  012760  055441  000024          MOV     #MR1AAA!MS!MCLK,RMMR1(R0)       ;LOAD RMMR1
3944
3945  013716  012760  051441  000024          MOV     #MR1AAA!MS,RMMR1(R0)    ;LOAD RMMR1
3946
3947  013724  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
3948  013732  042737  177737  001142          BIC     # CWC,$BDDAT
3949  013740  001013                          BNE     50$
3950  013742  005302                          DEC     R2
3951  013744  001361                          BNE     40$
3952  013746  012737  000040  001140          MOV     #WC,$GDDAT
3953  013754  010037  001136                  MOV     R0,$BDADR
3954  013760  062737  000024  001136          ADD     #RMMR1,$BDADR
3955  013766  104041                          ERROR   41                      ;DTE DIDNOT CLEAR WORD COUNT INH
3956  013770                          50$:
3957
3958
3959                                  ;;*****************************************************************
3960                                  ;*TEST 20        SYNC GENERATION TEST
3961
3962                                  ;;*****************************************************************
3963  013770  000004                  TST20:  SCOPE
3964  013772  012737  000020  001226          MOV     #20,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
3965  014000  000240                          NOP
3966  014002  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
3967  014010  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
3968  014016  012737  014032  001122          MOV     #T20,$LPADR  ;LOAD LOOP ON TEST ADDRESS
3969  014024  012737  014032  001124          MOV     #T20,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
3970  014032                          T20:
3971  014032  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
3972  014036  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
3973  014042  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
3974                                  ;SETUP REGISTER OUTPUT BUFFER FOR SUBROUTINES
3975  014046  012737  000000  001410          MOV     #0,RMDA0                ;SECTOR 0, TRACK 0
3976  014054  012737  000005  001436          MOV     #5,RMDC0                ;CYL 5
3977  014062  012737  010000  001434          MOV     #FMT16,RMOF0            ;16 BIT MODE
```

```
 3978  014070  012737  054062  001406          MOV     #BUFFER,RMBA0           ;BUFFER ADDRESS
 3979  014076  012737  177776  001404          MOV     # C2+1,RMWC0           ;WORD COUNT
 3980  014104  012737  000063  001402          MOV     #WH!GO,RMCS10           ;WHITE HEAD AND DATA COM
 3981
 3982                                  ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
 3983  014112  004737  023326                  JSR     PC,ENBSCH
 3984  014116  000403                          BR      10$             ;GO TO 10$ IF NO ERROR
 3985  014120  104000                          ERROR                   ;RETURN HERE IF ERROR
 3986                                  ;       BR      160$                    ;
 3987  014122  000137  014512                  JMP     160$                    ;
 3988  014126                          10$:
 3989                                  ;FORCE SECTOR COMPARE USING SUBROUTINE
 3990  014126  004737  024206                  JSR     PC,SCTCMP
 3991  014132  000402                          BR      20$             ;GO TO 20$ IF NO ERROR
 3992  014134  104000                          ERROR                   ;RETURN HERE IF ERROR
 3993  014136  000565                          BR      160$                    ;
 3994  014140                          20$:
 3995
 3996                                  ;;**************************************************************************
 3997                                  ;WRITE PROM OF DATA TIMING SEQUENCER SHOULD NOW BE ENABLED
 3998                                  ;FIRST, VERIFY THAT WRITE GATE IS ON, INDICATING THAT
 3999                                  ;SECTOR COMPARE SET
 4000
 4001
 4002  014140  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
 4003  014146  042737  177776  001142          BIC     # CBB00,$BDDAT          ;BUS BIT 0 IS WRITE GATE
 4004  014154  001011                          BNE     30$                     ;
 4005  014156  005037  001140                  CLR     $GDDAT
 4006  014162  010037  001136                  MOV     R0,$BDADR
 4007  014166  062737  000040  001136          ADD     #RMMR2,$BDADR
 4008  014174  104042                          ERROR   42                              ;WRITE GATE NOT ON  DURING
 4009  014176  000545                          BR      160$                    ;FORMAT-CS FAILURE
 4010  014200                          30$:
 4011                                  ;STEP THE DATA TIMING SEQUENCER TO LOCATION 16 AND VERIFY WRITE
 4012                                  ;GATE STAYS ON
 4013  014200  012702  000016                  MOV     #14.,R2                 ;MAXMUM NUMBER OF CLOCK
 4014  014204                          40$:
 4015
 4016  014204  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
 4017  014212  032737  000001  001142          BIT     #BB00,$BDDAT            ;
 4018  014220  001014                          BNE     50$                     ;BRANCH IF WRITE GATE ON
 4019  014222  005302                          DEC     R2
 4020  014224  001367                          BNE     40$
 4021  014226  010037  001136                  MOV     R0,$BDADR
 4022  014232  062737  000040  001136          ADD     #RMMR2,$BDADR
 4023  014240  012737  000001  001140          MOV     #BB00,$GDDAT
 4024  014246  104042                          ERROR   42                              ;WRITE GATE DROP DURING
 4025  014250  000520                          BR      160$                    ;FORMAT - CS FAILURE
 4026  014252                          50$:
 4027  014252                          60$:
 4028
 4029  014252  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4030
 4031  014260  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4032
 4033  014266  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
```

```
4034  014272  032703  000040            BIT     #WC,R3
4035  014276  001765                    BEQ     60$
4036                                ;PROM STROBE CAME ON-DECREMENT PROM STROBE COUNT
4037  014300  005302                    DEC     R2
4038  014302  001414                    BEQ     80$
4039                                ;WAIT FOR PROM STROBE TO GO OFF, THEN REPEAT LOOP
4040  014304                        70$:
4041
4042  014304  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4043
4044  014312  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4045
4046  014320  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4047  014324  032703  000040                  BIT     #WC,R3
4048  014330  001725                          BEQ     40$
4049  014332  000764                          BR      70$
4050  014334                        80$:
4051
4052                                ;;*****************************************************************
4053                                ;VERIFY HEADER SYNC GENERATION
4054                                ;WRITE DATA BIT IS INVERTED AT MAINTENANCE REGISTER
4055                                ;FIRST,COUNT NUMBER OF ZERO BITS UNTIL FIRST ONE BIT
4056  014334  012702  000020                  MOV     #16.,R2         ;MAX TIMES THRU LOOP
4057  014340  005003                          CLR     R3
4058  014342                        90$:
4059
4060  014342  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4061  014346  032704  000010                  BIT     #MWD,R4
4062  014352  001414                          BEQ     110$            ;JUMP OUT OF LOOP WITH FIRST 1
4063  014354  005203                          INC     R3              ;INCREMENT ZERO COUNT
4064  014356  005302                          DEC     R2              ;DECREMENT MAX LOOP COUNT
4065  014360  001002                          BNE     100$
4066  014362  104043                          ERROR   43
4067  014364  000452                          BR      160$            ;HEADER SYNC ,CANT GET FIRST 1
4068  014366                        100$:
4069
4070  014366  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4071
4072  014374  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4073  014402  000757                          BR      90$
4074  014404                        110$:
4075                                ;MAKE SURE THERE WERE AT LEAST 8 ZERO BITS IN HEADER SYNC
4076  014404  020327  000010                  CMP     R3,#8.
4077  014410  103002                          BHIS    120$
4078  014412  104043                          ERROR   43
4079  014414  000436                          BR      160$            ;HEADER SYNC
4080  014416                        120$:
4081                                ;SAMPLE AND STORE THE REST OF THE HEADER SYNC
4082  014416  012702  000010                  MOV     #8.,R2          ;NUMBER OF SAMPLES
4083  014422  005003                          CLR     R3              ;HEADER SYNC
4084  014424                        130$:
4085
4086  014424  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4087  014430  006303                          ASL     R3
4088  014432  032704  000010                  BIT     #MWD,R4         ;SHIFT SAMPLE,IF SYNC BIT IS 1
4089  014436  001002                          BNE     140$            ;THEN SET SAMPLE INPUT
```

```
 4090  014440  052703  000001            BIS     #BIT0,R3
 4091  014444  005302                140$:   DEC     R2
 4092  014446  001407                        BEQ     150$
 4093
 4094  014450  012760  055401  000024        MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4095
 4096  014456  012760  051401  000024        MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4097  014464  000757                        BR      130$            ;DO LOOP TIL R2=0
 4098  014466                        150$:
 4099                                ;VERIFY THE SYNC BIT STREAM IS 0000000010011000
 4100  014466  012737  000230  001140        MOV     #000230,$GDDAT
 4101  014474  010337  001142                MOV     R3,$BDDAT
 4102  014500  023737  001140  001142        CMP     $GDDAT,$BDDAT
 4103  014506  001401                        BEQ     160$
 4104  014510  104044                        ERROR   44                      ;INCORRECT HEADER SYNC
 4105  014512                        160$:
 4106                                ;;*****************************************************************
 4107                                ;*TEST 21        WRITE HEADER TEST
 4108
 4109                                ;;*****************************************************************
 4110  014512  000004                TST21:  SCOPE
 4111  014514  012737  000021  001226        MOV     #21,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
 4112  014522  000240                        NOP
 4113  014524  012737  000024  001120        MOV     #20.,$ICNT      ;20 ITERATIONS
 4114  014532  112737  000001  001131        MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
 4115  014540  012737  014554  001122        MOV     #T21,$LPADR  ;LOAD LOOP ON TEST ADDRESS
 4116  014546  012737  014554  001124        MOV     #T21,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
 4117  014554                        T21:
 4118  014554  012706  001100                MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4119  014560  013700  001276                MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
 4120  014564  013701  001456                MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
 4121  014570  012760  000040  000010        MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 4122  014576  111160  000010                MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
 4123                                ;SETUP THE REGISTER OUTPUT BUFFER FOR SUBROUTINES
 4124  014602  012737  002037  001410        MOV     #002037,RMDA0   ;TRACK 4, SECTOR 31
 4125  014610  012737  001466  001436        MOV     #1466,RMDC0     ;CYLINDER 822.
 4126  014616  012737  010000  001434        MOV     #FMT16,RMOF0    ;IN 16 BIT FORMAT
 4127  014624  012702  054062                MOV     #BUFFER,R2      ;BUFFER ADDRESS
 4128  014630  010237  001406                MOV     R2,RMBA0        ;BUFFER STARTS
 4129  014634  012722  151466                MOV     #151466,(R2)+   ;STORE FIRST HEADER WORD
 4130  014640  012712  002037                MOV     #002037,(R2)    ;STORE SECOND HEADER WORD
 4131  014644  012737  177776  001404        MOV     # C2+1,RMWC0     ;WORD CTR = 2
 4132  014652  012737  000063  001402        MOV     #WH!GO,RMCS10   ;WRITE HEADER COMMAND
 4133
 4134                                ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
 4135  014660  004737  023326                JSR     PC,ENBSCH
 4136  014664  000403                        BR      10$             ;GO TO 10$ IF NO ERROR
 4137  014666  104000                        ERROR                   ;RETURN HERE IF ERROR
 4138  014670  000137  015472                JMP     160$            ;
 4139  014674                        10$:
 4140                                ;FORCE SECTOR COMPARE USING SUBROUTINE
 4141  014674  004737  024206                JSR     PC,SCTCMP
 4142  014700  000403                        BR      20$             ;GO TO 20$ IF NO ERROR
 4143  014702  104000                        ERROR                   ;RETURN HERE IF ERROR
 4144  014704  000137  015472                JMP     160$
 4145  014710                        20$:
```

```
4146                                    ;;*****************************************************************
4147                                    ;STEP THE DATA TIMING SEQUENCER UNTIL "HEADER AREA" COMES ON
4148  014710  012702  000017                    MOV     #15.,R2         ;ALLOW 15 MORE PROM STORBES
4149  014714  012704  000041            30$:    MOV     #33.,R4         ;ALLOW 33 BIT CLOCKS PER PROM STROBE
4150
4151                                    ;PULSE BIT CLOCK UNTIL PROM STROBE IS ON
4152  014720                            40$:
4153
4154  014720  012760  055401  000024            MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4155
4156  014726  012760  051401  000024            MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4157
4158  014734  016003  000024                    MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4159  014740  032703  000040                    BIT     #WC,R3
4160  014744  001004                            BNE     50$
4161  014746  005304                            DEC     R4
4162  014750  001363                            BNE     40$
4163  014752  000137  015472                    JMP     160$            ;COUNT EXHAUSTED
4164  014756                            50$:
4165
4166                                    ;SEE IF HEADER AREA CAME ON
4167  014756  032703  000200                    BIT     #PHA,R3
4168  014762  001062                            BNE     80$
4169  014764  005302                            DEC     R2
4170  014766  001015                            BNE     60$
4171                                    ;ERROR-HEADER AREA NEVER CAME ON
4172  014770  012737  000200  001140            MOV     #PHA,$GDDAT     ;SETUP ERROR MESSAGE
4173  014776  005037  001142                    CLR     $BDDAT
4174  015002  010037  001136                    MOV     R0,$BDADR
4175  015006  062737  000024  001136            ADD     #RMMR1,$BDADR
4176  015014  104045                            ERROR   45                      ;HEADER AREA WONT SET
4177  015016  000137  015472                    JMP     160$            ;SKIP REST OF TEST
4178  015022                            60$:
4179                                    ;WAIT FOR PROM STROBE TO GO OFF
4180
4181  015022  012760  055401  000024            MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4182
4183  015030  012760  051401  000024            MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4184
4185  015036  016003  000024                    MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4186  015042  032703  000040                    BIT     #WC,R3
4187  015046  001404                            BEQ     70$
4188  015050  005304                            DEC     R4
4189  015052  001363                            BNE     60$
4190  015054  000137  015472                    JMP     160$            ;COUNT EXHAUSTED
4191  015060                            70$:
4192                                    ;VERIFY THE TAG BUS ONCE EVERY PROM STROBE
4193
4194  015060  016037  000040  001142            MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
4195  015066  042737  176000  001142            BIC     # C1777,$BDDAT
4196  015074  022737  000001  001142            CMP     #BB00,$BDDAT    ;WRITE GATE SHOULD BE ON
4197  015102  001704                            BEQ     30$             ;ALL ELSE OFF
4198  015104  010037  001136                    MOV     R0,$BDADR       ;
4199  015110  062737  000040  001136            ADD     #RMMR2,$BDADR
4200  015116  012737  000001  001140            MOV     #BB00,$GDDAT
4201  015124  104042                            ERROR   42                      ;TAG BUS WRONG DURING FORMAT
```

```
4202  015126  000561                          BR      160$
4203  015130                          80$:
4204                                  ;HEADER AREA IS NOW ON CAN START SAMPLE
4205                                  ;DATA AFTER 5 MORE BIT CLOCKS
4206  015130  012702  000005                  MOV     #5,R2
4207  015134                          81$:
4208
4209  015134  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4210
4211  015142  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4212  015150  005302                          DEC     R2
4213  015152  001370                          BNE     81$
4214                                  ;;*****************************************************************
4215                                  ;SAMPLE AND STORE THE TWO HEADER WORDS
4216  015154  005037  001174                  CLR     $TMP0
4217  015160  005037  001176                  CLR     $TMP1
4218  015164  012702  001174                  MOV     #$TMP0,R2       ;R2 NUMBER OF WORDS/ADDRESS
4219  015170  012703  000020          90$:    MOV     #16.,R3         ;NUMBER OF BITS
4220  015174  005004                          CLR     R4              ;WORD STORAGE
4221  015176                          100$:
4222
4223  015176  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
4224  015202  006304                          ASL     R4              ;SHIFT WORD SAMPLE
4225  015204  032705  000010                  BIT     #MWD,R5         ;SET BIT 0 IF WRITE BIT ON
4226  015210  001002                          BNE     110$
4227  015212  052704  000001                  BIS     #BIT0,R4
4228  015216                          110$:
4229
4230  015216  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4231
4232  015224  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4233  015232  005303                          DEC     R3
4234  015234  001360                          BNE     100$
4235  015236  010422                          MOV     R4,(R2)+        ;STORE WORD SAMPLE
4236  015240  022702  001200                  CMP     #$TMP0+4,R2     ;DONE ?
4237  015244  101415                          BLOS    120$            ;YES
4238                                  ;HEADER AREA  SHOULD STILL BE ON FOR SECOND HEADER WORD
4239  015246  042705  177577                  BIC     # C<PHA>,R5
4240  015252  001346                          BNE     90$             ;BRANCH IF ON
4241  015254  010037  001136                  MOV     R0,$BDADR
4242  015260  062737  000024  001136          ADD     #RMMR1,$BDADR
4243  015266  012737  000200  001140          MOV     #PHA,$GDDAT
4244  015274  104045                          ERROR   45
4245  015276  000475                          BR      160$
4246  015300                          120$:
4247                                  ;;*****************************************************************
4248                                  ;WAIT UNTIL ECRC SETS
4249  015300  005003                          CLR     R3              ;CHECK NUMBER BITS DIFFER BET PHA AND ECRC
4250
4251  015302                          125$:
4252
4253  015302  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
4254  015306  042705  176777                  BIC     # C<ECRC>,R5    ;CHECK OUT THE REST BITS
4255  015312  001026                          BNE     127$            ;BRANCH IF ECRC ON
4256
4257  015314  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
```

```
4258
4259  015322  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4260  015330  005203                          INC     R3                      ;TOTAL NUMBER OF BIT DIFF
4261  015332  022703  000020                  CMP     #16.,R3                 ;OVER ONE WORD ?
4262  015336  101361                          BHI     125$                    ;BRANCH IF NOT
4263  015340  010037  001136                  MOV     R0,$BDADR
4264  015344  062737  000024  001136          ADD     #RMMR1,$BDADR
4265  015352  012737  001000  001140          MOV     #ECRC,$GDDAT
4266  015360  005037  001142                  CLR     $BDDAT                  ;CLEAR THE EXP
4267  015364  104046                          ERROR   46
4268  015366  000441                          BR      160$                    ;EXIT
4269  015370                          127$:
4270                                  ;;**************************************************************
4271                                  ;SAMPLE AND STORE CRC WORD
4272  015370  012703  000020                  MOV     #16.,R3
4273  015374  005004                          CLR     R4
4274  015376                          130$:
4275
4276  015376  016005  000024                  MOV     RMMR1(R0),R5            ;STORE RMMR1 AT R5
4277  015402  006304                          ASL     R4                      ;SHIFT WORD SAMPLE
4278  015404  032705  000010                  BIT     #MWD,R5                 ;SET BIT 0 IF WRITE BIT ON
4279  015410  001002                          BNE     140$
4280  015412  052704  000001                  BIS     #BIT0,R4
4281  015416                          140$:
4282
4283  015416  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4284
4285  015424  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4286  015432  005303                          DEC     R3
4287  015434  001360                          BNE     130$
4288  015436  010422                          MOV     R4,(R2)+
4289                                  ;;**************************************************************
4290                                  ;VERIFY 3 HEADER WORDS
4291                                  ;NOTE THAT DATA WAS STORED IN THE REVERSE
4292                                  ;ORDER FROM WHICH IT WAS WRITTEN
4293  015440  022737  066313  001174          CMP     #066313,$TMP0
4294  015446  001010                          BNE     150$                    ;BRANCH IF ERROR
4295  015450  022737  174040  001176          CMP     #174040,$TMP1
4296  015456  001004                          BNE     150$                    ;BRANCH IF ERROR
4297  015460  022737  167501  001200          CMP     #167501,$TMP2           ;
4298  015466  001401                          BEQ     160$                    ;BRANCH IF NO ERROR
4299  015470                          150$:
4300  015470  104047                          ERROR   47
4301  015472                          160$:                                   ;END OF SUB TEST
4302                                  ;;**************************************************************
4303                                  ;*TEST 22        HEADER COMPARE TEST
4304
4305                                  ;;**************************************************************
4306  015472  000004                  TST22:  SCOPE
4307  015474  012737  000022  001226          MOV     #22,$TESTN              ;;SET TEST NUMBER IN APT MAIL BOX
4308  015502  000240                          NOP
4309  015504  012737  000024  001120          MOV     #20.,$ICNT              ;20 ITERATIONS
4310  015512  112737  000001  001131          MOVB    #1,$ERMAX               ;ONE ERROR ALLOWED
4311  015520  012737  015534  001122          MOV     #T22,$LPADR     ;LOAD LOOP ON TEST ADDRESS
4312  015526  012737  015534  001124          MOV     #T22,$LPERR     ;LOAD LOOP ON ERROR ADDRESS
4313  015534                          T22:
```

```
 4314  015534  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
 4315  015540  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
 4316  015544  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
 4317
 4318                                  ;LOAD REGISTER OUTPUT BUFFER FOR READ HEADER COMMAND
 4319
 4320  015550  012737  001466  001436          MOV     #001466,RMDC0           ;CYL=822.
 4321  015556  012737  002037  001410          MOV     #2037,RMDA0             ;TRK=4,SEC=31.
 4322  015564  012737  054062  001406          MOV     #BUFFER,RMBA0           ;DATA ADDRESS
 4323  015572  012737  177776  001404          MOV     # C2+1,RMWC0           ;READ HEADER ONLY
 4324  015600  012737  010000  001434          MOV     #FMT16,RMOF0            ;16 BIT MODE
 4325  015606  012737  000073  001402          MOV     #RH!GO,RMCS10           ;READ HEADER AND DATA FUN
 4326
 4327                                  ;CLEAR THE MASSBUS
 4328  015614  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 4329  015622  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
 4330
 4331                                  ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
 4332  015626  004737  023326                  JSR     PC,ENBSCH
 4333  015632  000403                          BR      10$             ;GO TO 10$ IF NO ERROR
 4334  015634  104000                          ERROR                   ;RETURN HERE IF ERROR
 4335  015636  000137  016742                  JMP     230$            ;SKIP REST OF TEST IF ERROR
 4336  015642                          10$:
 4337                                  ;FORCE SECTOR COMPARE USING SUBROUTINE
 4338  015642  004737  024206                  JSR     PC,SCTCMP
 4339  015646  000403                          BR      20$             ;GO TO 20$ IF NO ERROR
 4340  015650  104000                          ERROR                   ;RETURN HERE IF ERROR
 4341  015652  000137  016742                  JMP     230$
 4342  015656                          20$:
 4343                                  ;READ GATE SHOULD BE OFF FOR 3 PROM STROBES
 4344
 4345  015656  010037  001136                  MOV     R0,$BDADR               ;SETUP ERROR MESSAGE
 4346  015662  062737  000040  001136          ADD     #RMMR2,$BDADR
 4347  015670  012702  000004                  MOV     #4,R2                   ;R2=NUMBER OF PROM STROBES
 4348  015674  012704  000021          30$:    MOV     #17.,R4         ;MAX BIT CLOCKS
 4349
 4350  015700                          40$:
 4351
 4352                                  ;CLOCK BIT CLOCK UNTIL PROM STROBE COMES ON
 4353
 4354
 4355  015700  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4356
 4357  015706  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4358
 4359  015714  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
 4360  015720  032703  000040                  BIT     #WC,R3
 4361  015724  001004                          BNE     50$             ;BRANCH WHEN PROM STROBE ON
 4362  015726  005304                          DEC     R4
 4363  015730  001363                          BNE     40$
 4364  015732  000137  016742                  JMP     230$            ;***********
 4365  015736                          50$:
 4366
 4367                                  ;VERIFY READ GATE IS OFF AND TAG BUS IS ZERO
 4368
 4369
```

```
4370  015736  016003  000040          MOV     RMMR2(R0),R3    ;STORE RMMR2 AT R3
4371  015742  042703  176000          BIC     # C1777,R3
4372  015746  001407                  BEQ     60$             ;BRANCH IF TAG BUS ZERO
4373  015750  010337  001142          MOV     R3,$BDDAT
4374  015754  005037  001140          CLR     $GDDAT
4375  015760  104050                  ERROR   50
4376  015762  000137  016742          JMP     230$            ;READ HEADER FUNCTION
4377  015766                  60$:
4378                          ;CLOCK BIT CLOCK TIL PROM STROBE GOES OFF
4379
4380
4381  015766  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4382
4383  015774  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4384
4385  016002  016003  000024          MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4386  016006  032703  000040          BIT     #WC,R3
4387  016012  001404                  BEQ     70$             ;BRANCH IF PROM STROBE OFF
4388  016014  005304                  DEC     R4
4389  016016  001363                  BNE     60$
4390  016020  000137  016742          JMP     230$            ;***************
4391  016024                  70$:
4392                          ;CONTINUE FOR 3 BIT CYCLE TOTAL (LOC 0 NOT SAMPLED)
4393
4394  016024  005302                  DEC     R2
4395  016026  001322                  BNE     30$
4396
4397                          ;;*****************************************************************
4398                          ;READ GATE SHOULD COME ON WITH NEXT PROM STROBE AND STAY
4399                          ;ON FOR 7 CYCLES (AND MORE)
4400
4401  016030  012702  000006          MOV     #6,R2
4402  016034  012703  000021  80$:    MOV     #17.,R3
4403                          ;CLOCK BIT CLOCK TIL PROM STROBE SETS
4404
4405  016040                  90$:
4406
4407  016040  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4408
4409  016046  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4410
4411  016054  016004  000024          MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4412  016060  032704  000040          BIT     #WC,R4
4413  016064  001004                  BNE     100$            ;BRANCH WHEN PROM STROBE ON
4414  016066  005303                  DEC     R3
4415  016070  001363                  BNE     90$
4416  016072  000137  016742          JMP     230$            ;************
4417  016076                  100$:
4418                          ;VERIFY READ GATE ON AND REST OF TAG BUS OFF
4419
4420  016076  016004  000040          MOV     RMMR2(R0),R4    ;STORE RMMR2 AT R4
4421  016102  042704  176000          BIC     # C1777,R4
4422  016106  022704  000002          CMP     #BB01,R4
4423  016112  001410                  BEQ     110$            ;BRANCH IF READ GATE ON
4424  016114  010437  001142          MOV     R4,$BDDAT
4425  016120  012737  000002  001140          MOV     #BB01,$GDDAT
```

```
4426  016126  104050                  ERROR   50                      ;TAG BUS INCORRECT DURING
4427  016130  000137  016742          JMP     230$            ;READ HEADER FUNCTION
4428
4429  016134                  110$:
4430                          ;CLOCK BIT CLOCK TIL PROM STROBE GOES OFF
4431
4432
4433  016134  012760  055401  000024  MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4434
4435  016142  012760  051401  000024  MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4436
4437  016150  016004  000024          MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
4438  016154  032704  000040          BIT     #WC,R4
4439  016160  001404                  BEQ     120$            ;BRANCH WHEN PROM STROBE OFF
4440  016162  005303                  DEC     R3
4441  016164  001363                  BNE     110$
4442  016166  000137  016742          JMP     230$
4443
4444  016172                  120$:
4445                          ;CONTINUE FOR TOATL OF 7 CYCLES
4446  016172  005302                  DEC     R2
4447  016174  001317                  BNE     80$     ;*&**&&
4448                          ;;*****************************************************************
4449                          ;LOOKING FOR SYNC SHOULD SET WITH NEXT PROM STORBE
4450  016176  012702  000021          MOV     #17.,R2         ;MAX BIT CLOCKS
4451  016202                  130$:
4452                          ;CLOCK BIT CLOCK TIL PROM STROBE SETS
4453
4454
4455  016202  012760  055401  000024  MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4456
4457  016210  012760  051401  000024  MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4458
4459  016216  016003  000024          MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4460  016222  032703  000040          BIT     #WC,R3
4461  016226  001004                  BNE     140$            ;BRANCH WHEN PROM STROBE ON
4462  016230  005302                  DEC     R2
4463  016232  001363                  BNE     130$
4464  016234  000137  016742          JMP     230$
4465  016240                  140$:
4466                          ;VERIFY "PLFS" IS NOW ON
4467  016240  042703  175777          BIC     # CPLFS,R3
4468  016244  001015                  BNE     150$
4469  016246  010337  001142          MOV     R3,$BDDAT
4470  016252  012737  002000  001140  MOV     #PLFS,$GDDAT
4471  016260  010037  001136          MOV     R0,$BDADR
4472  016264  062737  000024  001136  ADD     #RMMR1,$BDADR
4473  016272  104035                  ERROR   35
4474  016274  000137  016742          JMP     230$
4475  016300                  150$:
4476  016300  012703  000005          MOV     #5,R3
4477                          ;WITH PLFS ON, WORD COUNT INHIBIT WILL SET IN 5 BIT CLOCKS
4478  016304                  155$:
4479
4480  016304  012760  055401  000024  MOV     #MR1AAA.MCLK,RMMR1(R0)  ;LOAD RMMR1
4481
```

```
4482  016312  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4483  016320  005303                          DEC     R3
4484  016322  001370                          BNE     155$
4485
4486                                  ;;*****************************************************************
4487                                  ;SIMULATE THE SYNC PATTERN BEING READ
4488  016324  012702  000031                  MOV     #31,R2          ;SYNC PATTERN=00011001
4489  016330  012703  000010                  MOV     #8.,R3          ;8 BITS IN PATTERN
4490  016334                          160$:
4491  016334  012737  055401  001426          MOV     #MR1AAA!MCLK,RMMR10
4492  016342  000241                          CLC
4493  016344  006002                          ROR     R2
4494  016346  103003                          BCC     165$
4495  016350  052737  002000  001426          BIS     #MRD,RMMR10
4496  016356                          165$:
4497
4498  016356  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4499  016364  042737  004000  001426          BIC     #MCLK,RMMR10
4500
4501  016372  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4502  016400  005303                          DEC     R3              ;CONTINUE TIL SHIFT COUNT
4503  016402  001354                          BNE     160$            ;IS ZERO
4504                                  ;;*****************************************************************
4505                                  ;SIMULATE THE HEADER BEING READ
4506  016404  012702  001174                  MOV     #$TMP0,R2
4507  016410  012737  151466  001174          MOV     #151466,$TMP0
4508  016416  012737  002037  001176          MOV     #2037,$TMP1     ;*************
4509  016424  012737  101367  001200          MOV     #101367,$TMP2           ;CRC PATTERN ***
4510  016432                          170$:
4511  016432  012703  000020                  MOV     #16.,R3         ;NUMBER OF BITS EACH WORD
4512  016436  012204                          MOV     (R2)+,R4        ;HEADER WORD 1,2 OR 3
4513  016440                          175$:
4514  016440  012737  055401  001426          MOV     #MR1AAA!MCLK,RMMR10
4515  016446  000241                          CLC
4516  016450  006004                          ROR     R4
4517  016452  103003                          BCC     180$
4518  016454  052737  002000  001426          BIS     #MRD,RMMR10
4519  016462                          180$:
4520
4521  016462  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4522  016470  042737  004000  001426          BIC     #MCLK,RMMR10
4523
4524  016476  013760  001426  000024          MOV     RMMR10,RMMR1(R0)        ;LOAD RMMR1
4525  016504  005303                          DEC     R3              ;SHIFT OUT 16 BITS
4526  016506  001354                          BNE     175$
4527  016510  020227  001200                  CMP     R2,#$TMP0+4     ;ALL DONE ?
4528  016514  101746                          BLOS    170$            ;BRANCH IF NOT
4529                                  ;;*****************************************************************
4530                                  ;LOOKING FOR SYNC SHOULD COME ON WITHIN 6 STROBES
4531  016516  012702  000006                  MOV     #6,R2
4532
4533  016522  012703  000021          185$:   MOV     #17.,R3
4534                                  ;CLOCK UNTIL PROM STROBE ON
4535  016526                          190$:
4536
4537  016526  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ,LOAD RMMR1
```

```
 4538
 4539  016534  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4540
 4541  016542  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
 4542  016546  032704  000040                  BIT     #WC,R4
 4543  016552  001003                          BNE     195$
 4544  016554  005303                          DEC     R3
 4545  016556  001363                          BNE     190$
 4546  016560  000470                          BR      230$
 4547  016562                          195$:
 4548                                  ;SEE IF "PLFS" IS ON
 4549  016562  042704  175777                  BIC     # CPLFS,R4
 4550  016566  001034                          BNE     210$
 4551                                  ;CONTINUE IF LESS THAN 6 PROM STROBES
 4552  016570  005302                          DEC     R2
 4553  016572  001014                          BNE     200$
 4554  016574  010437  001142                  MOV     R4,$BDDAT       ;SETUP ERROR
 4555  016600  012737  002000  001140          MOV     #PLFS,$GDDAT
 4556  016606  010037  001136                  MOV     R0,$BDADR
 4557  016612  062737  000024  001136          ADD     #RMMR1,$BDADR
 4558  016620  104035                          ERROR   35
 4559  016622  000447                          BR      230$            ;HEADER
 4560  016624                          200$:
 4561                                  ;CLOCK UNTIL PROM STROBE OFF
 4562
 4563  016624  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4564
 4565  016632  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4566
 4567  016640  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
 4568  016644  032704  000040                  BIT     #WC,R4
 4569  016650  001724                          BEQ     185$
 4570  016652  005303                          DEC     R3
 4571  016654  001363                          BNE     200$
 4572  016656  000431                          BR      230$
 4573  016660                          210$:
 4574                                  ;VERIFY NO HEADER ERROR IS SET
 4575
 4576  016660  016037  000014  001142          MOV     RMER1(R0),$BDDAT        ;STORE RMER1 AT $BDDAT
 4577  016666  042737  177157  001142          BIC     # C<HCRC!HCE!FER>,$BDDAT
 4578  016674  001411                          BEQ     220$
 4579  016676  005037  001140                  CLR     $GDDAT
 4580  016702  010037  001136                  MOV     R0,$BDADR
 4581  016706  062737  000014  001136          ADD     #RMER1,$BDADR
 4582  016714  104051                          ERROR   51
 4583  016716  000411                          BR      230$            ;HEADER ERROR SET
 4584  016720                          220$:
 4585                                  ;VERIFY DATA IN MEMORY OK
 4586  016720  023737  054062  001174          CMP     BUFFER,$TMP0    ;COMPARE 1 ST HEADER WORD
 4587  016726  001004                          BNE     225$
 4588  016730  023737  054064  001176          CMP     BUFFER+2,$TMP1  ;COMPARE 2 ND HEADER WORD
 4589  016736  001401                          BEQ     230$
 4590  016740                          225$:
 4591                                  ;REPORT ERROR IN ONE OR MORE HEADER WORDS
 4592  016740  104052                          ERROR   52
 4593  016742                          230$:
```

```
4594                                    ;;**********************************************************************
4595                                    ;*TEST 23       ECC GENERATION TEST
4596                                    ;
4597                                    ;;**********************************************************************
4598  016742  000004                    TST23:  SCOPE
4599  016744  012737  000023  001226            MOV     #23,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
4600  016752  000240                            NOP
4601  016754  012737  000024  001120            MOV     #20.,$ICNT      ;20 ITERATIONS
4602  016762  112737  000001  001131            MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
4603  016770  012737  017004  001122            MOV     #T23,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4604  016776  012737  017004  001124            MOV     #T23,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4605  017004                            T23:
4606  017004  012706  001100                    MOV     #STACK,SP       ;LOAD THE STACK POINTER
4607  017010  013700  001276                    MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
4608  017014  013701  001456                    MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
4609  017020  012760  000040  000010            MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
4610  017026  111160  000010                    MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
4611                                    ;SETUP REGISTER OUTPUT BUFFER FOR WRITE DATA CAMOMMAND
4612  017032  012737  002037  001410            MOV     #002037,RMDA0   ;TRK 4, SEC 31
4613  017040  012737  001466  001436            MOV     #1466,RMDC0     ;CYL=822.
4614  017046  012737  012000  001434            MOV     #FMT16!HCI,RMOF0        ;16 BIT FORMAT,IGNORE HEADER
4615  017054  012702  054062                    MOV     #BUFFER,R2
4616  017060  010237  001406                    MOV     R2,RMBA0                ;DATA ADDRESS
4617  017064  012703  177400                    MOV     # C256.+1,R3
4618  017070  010337  001404                    MOV     R3,RMWC0
4619  017074  012737  000061  001402            MOV     #WD!GO,RMCS10           ;WRITE DATA COMMAND ***
4620                                    ;FILL THE DATA BUFFER WITH ALL ONES
4621  017102  012704  177777                    MOV     #177777,R4              ;DATA PATTERN ALL ONES
4622  017106  012703  000400                    MOV     #400,R3         ;ONE SECTOR DATA FIELD
4623  017112  010422                    10$:    MOV     R4,(R2)+
4624  017114  005303                            DEC     R3
4625  017116  001375                            BNE     10$
4626
4627                                    ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
4628  017120  004737  023326                    JSR     PC,ENBSCH
4629  017124  000403                            BR      20$             ;GO TO 20$ IF NO ERROR
4630  017126  104000                            ERROR                   ;RETURN HERE IF ERROR
4631  017130  000137  017726                    JMP     220$
4632  017134                            20$:
4633                                    ;FORCE SECTOR COMPARE USING SUBROUTINE
4634  017134  004737  024206                    JSR     PC,SCTCMP
4635  017140  000403                            BR      30$             ;GO TO 30$ IF NO ERROR
4636  017142  104000                            ERROR                   ;RETURN HERE IF ERROR
4637  017144  000137  017726                    JMP     220$
4638  017150                            30$:
4639                                    ;SET "LOOKING FOR SYNC" USING SUBROUTINE
4640  017150  004737  024314                    JSR     PC,SETLFS
4641  017154  000403                            BR      40$
4642  017156  104000                            ERROR
4643  017160  000137  017726                    JMP     220$
4644  017164                            40$:
4645                                    ;CLOCK THE SYNC PATTERN USING SUBROUTINE
4646  017164  004737  024476                    JSR     PC,CLKSNC
4647  017170  000403                            BR      50$
4648  017172  104000                            ERROR
4649  017174  000137  017726                    JMP     220$
```

```
                                ;HEADER COMPARE IS INHIBITED FOR THIS TEST
                                ;CLOCK THE DATA TIMING SEQUENCER AND VERIFY THE FOLLOWING WAVEFORMS
                                ;FOR EACH LEADING EDGD OF PROM STROBE
                                ;       HEADER ARES, READ GATE (DATA TIMING SEQUENCER=200 OCTAL)
                                ;       ENABLE CRC OUT
                                ;       WRITE GATE

017200  012702  017732          50$:    MOV     #230$,R2        ;****** POINTER TO WAVE FORM TABLE

017204                          70$:
                                ;CLOCK BIT CLOCK (MCLK) UNTIL PROM STROBE COMES ON
017204  004737  017752                  JSR     PC,300$
017210                          80$:
                                ;PROM STROBE IS ON - VERIFY WAVEFORM

017210  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
017216  042737  176157  001142          BIC     # C<ECRC!PDA!PHA!EECC>,$BDDAT
017224  012237  001140                  MOV     (R2)+,$GDDAT
017230  023737  001140  001142          CMP     $GDDAT,$BDDAT
017236  001410                          BEQ     90$             ;BRANCH IF RMMR1 OK
                                ;ERROR - THE DATA TIMING SEQUENCER OUTPUT IS ONCORRECT
017240  010037  001136                  MOV     R0,$BDADR
017244  062737  000024  001136          ADD     #RMMR1,$BDADR
017252  104054                          ERROR   54                      ;DATA TIMING SEQUENCER WRONG
017254  000137  017726                  JMP     220$            ;
017260                          90$:
                                ;VERIFY THE TAB BUS

017260  016037  000040  001142          MOV     RMMR2(R0),$BDDAT        ;STORE RMMR2 AT $BDDAT
017266  042737  176000  001142          BIC     # C1777,$BDDAT
017274  012237  001140                  MOV     (R2)+,$GDDAT
017300  023737  001140  001142          CMP     $GDDAT,$BDDAT
017306  001410                          BEQ     100$            ;BRANCH IF TAG BUS OK
                                ;ERROR- TAG BUS IS WRONG
017310  010037  001136                  MOV     R0,$BDADR
017314  062737  000040  001136          ADD     #RMMR2,$BDADR
017322  104053                          ERROR   53                      ;TAG BUS WRONG
017324  000137  017726                  JMP     220$
017330                          100$:
                                ;CLOCK BIT CLOCK TIL PROM STROBE RESETS
017330  004737  020060                  JSR     PC,350$
017334                          110$:
                                ;CONTINUE CHECKING THROUGH 4 PROM CYCLES
017334  020227  017750                  CMP     R2,#230$+16     ;********
017340  103721                          BLO     70$
                                ;;*****************************************************************
                                ;VERIFY THAT DATA AREA COMES ON WITHIN  9 PROM STROBES
017342  012702  000011                  MOV     #9.,R2          ;R2=9 STROBES
017346                          130$:
                                ;CLOCK PROM STROBE ON
017346  004737  017752                  JSR     PC,300$
017352                          140$:
                                ;PROM STROBE ON CKECK IF DATA AREA ON

017352  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
```

```
 4706  017356  042704  177377                  BIC     # CPDA,R4
 4707  017362  001020                          BNE     160$
 4708                                  ;CLOCK PROM STROBE OFF
 4709
 4710  017364                                  150$:
 4711  017364  004737  020060                  JSR     PC,350$
 4712  017370  005302                          DEC     R2
 4713  017372  001365                          BNE     130$
 4714                                  ;ERROR-DATA AREA DIDN'T COME ON
 4715  017374  012737  000400  001140          MOV     #PDA,$GDDAT     ;SETUP ERROR MESSAGE
 4716  017402  005037  001142                  CLR     $BDDAT
 4717  017406  010037  001136                  MOV     R0,$BDADR
 4718  017412  062737  000024  001136          ADD     #RMMR1,$BDADR
 4719  017420  104055                          ERROR   55
 4720  017422  000541                          BR      220$
 4721  017424                                  160$:
 4722                                  ;;*****************************************************************
 4723                                  ;VERIFY THAT DATA AREA IS ON FOR 256 CYCLES
 4724  017424  012702  000400                  MOV     #256.,R2
 4725  017430                                  170$:
 4726                                  ;VERIFY THAT DATA AREA IS ON AND ECC IS OFF
 4727
 4728  017430  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
 4729  017434  042703  177357                  BIC     # C<PDA!EECC>,R3
 4730  017440  022703  000400                  CMP     #PDA,R3
 4731  017444  001414                          BEQ     180$            ;DATA AREA IS ON
 4732  017446  010337  001142                  MOV     R3,$BDDAT
 4733  017452  012737  000400  001140          MOV     #PDA,$GDDAT
 4734  017460  010037  001136                  MOV     R0,$BDADR
 4735  017464  062737  000024  001136          ADD     #RMMR1,$BDADR
 4736  017472  104055                          ERROR   55                      ;DATA ERROR NOT ON
 4737  017474  000514                          BR      220$
 4738  017476                                  180$:
 4739                                  ;CLOCK PROM STROBE OFF
 4740  017476  004737  020060                  JSR     PC,350$
 4741                                  ;CLOCK PROM STROBE ON
 4742  017502  004737  017752                  JSR     PC,300$
 4743                                  ;CONTINUE TIL COUNT ZERO
 4744  017506  005302                          DEC     R2
 4745  017510  001347                          BNE     170$
 4746                                  ;ECC SHOULD  BE ENABLED WITHIN 4 CLOCK BITS
 4747
 4748  017512  005002                          CLR     R2
 4749  017514                                  182$:
 4750
 4751  017514  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
 4752  017520  042703  177357                  BIC     # C<PDA!EECC>,R3
 4753  017524  022703  000020                  CMP     #EECC,R3
 4754  017530  001426                          BEQ     190$
 4755  017532  005202                          INC     R2
 4756
 4757  017534  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4758
 4759  017542  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4760  017550  022702  000007                  CMP     #7,R2
 4761  017554  101357                          BHI     182$
```

```
 4762  017556  010337  001142                  MOV     R3,$BDDAT
 4763  017562  012737  000020  001140          MOV     #EECC,$GDDAT
 4764  017570  010037  001136                  MOV     R0,$BDADR
 4765  017574  062737  000024  001136          ADD     #RMMR1,$BDADR
 4766  017602  104056                          ERROR   56
 4767  017604  000450                          BR      220$
 4768  017606                          190$:
 4769                                  ;SAMPLE ECC CHARACTER FOR 32 BITS
 4770  017606  012702  001174                  MOV     #$TMP0,R2
 4771  017612  005003                  195$:   CLR     R3
 4772  017614  012704  000020                  MOV     #16.,R4
 4773  017620                          200$:
 4774
 4775  017620  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
 4776  017624  006303                          ASL     R3
 4777  017626  032705  000010                  BIT     #MWD,R5
 4778  017632  001002                          BNE     205$            ;BRANCH IF ECC BIT OFF (MWD INVERTED)
 4779  017634  052703  000001                  BIS     #BIT0,R3
 4780  017640                          205$:
 4781
 4782  017640  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4783
 4784  017646  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 4785  017654  005304                          DEC     R4
 4786  017656  001360                          BNE     200$
 4787  017660  010322                          MOV     R3,(R2)+
 4788  017662  022702  001200                  CMP     #$TMP0+4,R2
 4789  017666  101351                          BHI     195$
 4790                                  ;;*************************************************************
 4791                                  ;VERIFY ECC CHARACTER GENERATED
 4792  017670  022737  131177  001174          CMP     #131177,$TMP0   ;********
 4793  017676  001004                          BNE     210$
 4794  017700  022737  175154  001176          CMP     #175154,$TMP1   ;*****
 4795  017706  001407                          BEQ     220$
 4796  017710  012737  131177  001200  210$:   MOV     #131177,$TMP2   ;******
 4797  017716  012737  175154  001202          MOV     #175154,$TMP3   ;******
 4798  017724  104061                          ERROR   61                      ;ECC GENERATED INCORRECT
 4799  017726                          220$:
 4800  017726  000137  020164                  JMP     400$            ;GREAT ESCAPE
 4801
 4802  017732  000200                  230$:   .WORD   PHA
 4803  017734  000002                          .WORD   BB01
 4804  017736  000200                          .WORD   PHA
 4805  017740  000002                          .WORD   BB01
 4806  017742  000000                          .WORD   0
 4807  017744  000002                          .WORD   2               ;*******
 4808  017746  000000                          .WORD   0               ;******
 4809  017750  000000                          .WORD   0               ;
 4810
 4811
 4812  017752  012737  000021  020054  300$:   MOV     #17.,320$
 4813                                  ;                                       CLOCK PROM STROBE ON
 4814  017760                          305$:
 4815
 4816  017760  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 4817
```

```
4818  017766  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4819
4820  017774  016037  000024  020056          MOV     RMMR1(R0),330$  ;STORE RMMR1 AT 330$
4821  020002  042737  177737  020056          BIC     # CWC,330$
4822  020010  001020                          BNE     310$            ;EXIT IF PROM STROBE ON
4823  020012  005337  020054                  DEC     320$
4824  020016  001360                          BNE     305$
4825  020020  013737  020054  001142          MOV     320$,$BDDAT     ;SETUP ERROR MESSAGE
4826  020026  012737  000040  001140          MOV     #WC,$GDDAT
4827  020034  010037  001136                  MOV     R0,$BDADR
4828  020040  062737  000024  001136          ADD     #RMMR1,$BDADR
4829  020046  104030                          ERROR   30
4830  020050  000445                          BR      400$            ;
4831  020052  000207                  310$:   RTS     PC
4832  020054  000000                  320$:   .WORD   0               ;MAX BIT COUNT
4833  020056  000000                  330$:   .WORD   0               ;RMMR1 CONTENTS
4834
4835  020060  012737  000021  020160  350$:   MOV     #17.,370$
4836                                                                  ;CLOCK PROM STROBE OFF
4837  020066                          355$:
4838
4839  020066  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
4840
4841  020074  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
4842
4843  020102  016037  000024  020162          MOV     RMMR1(R0),380$  ;STORE RMMR1 AT 380$
4844  020110  042737  177737  020162          BIC     # CWC,380$
4845  020116  001417                          BEQ     360$
4846  020120  005337  020160                  DEC     370$
4847  020124  001360                          BNE     355$
4848  020126  013737  020162  001142          MOV     380$,$BDDAT
4849  020134  005037  001140                  CLR     $GDDAT
4850  020140  010037  001136                  MOV     R0,$BDADR
4851  020144  062737  000024  001136          ADD     #RMMR1,$BDADR
4852  020152  104062                          ERROR   62                      ;CAN'T CLEAR PROM STROBE
4853  020154  000403                          BR      400$
4854  020156  000207                  360$:   RTS     PC
4855
4856  020160  000000                  370$:   .WORD   0               ;MAX BIT COUNT
4857  020162  000000                  380$:   .WORD   0               ;CONTENTS OF RMMR1
4858
4859  020164                          400$:                           ;SUB-TEST EXIT POINT
4860  020164  000240                          NOP                     ;****
4861                                  ;;**************************************************************
4862                                  ;*TEST 24        ECC DETECTION TEST
4863
4864                                  ;;**************************************************************
4865  020166  000004                  TST24:  SCOPE
4866  020170  012737  000024  001226          MOV     #24,$TESTN      ;;SET TEST NUMBER IN APT MAIL BOX
4867  020176  000240                          NOP
4868  020200  012737  000024  001120          MOV     #20.,$ICNT      ;20 ITERATIONS
4869  020206  112737  000001  001131          MOVB    #1,$ERMAX       ;ONE ERROR ALLOWED
4870  020214  012737  020230  001122          MOV     #T24,$LPADR  ;LOAD LOOP ON TEST ADDRESS
4871  020222  012737  020230  001124          MOV     #T24,$LPERR  ;LOAD LOOP ON ERROR ADDRESS
4872  020230                          T24:
4873  020230  012706  001100                  MOV     #STACK,SP       ;LOAD THE STACK POINTER
```

```
4874  020234  013700  001276                  MOV     $BASE,R0        ;R0 = UNIBUS ADDRESS OF UUT
4875  020240  013701  001456                  MOV     TSTQUE,R1       ;R1 = POINTER TO DEVICE
4876  020244  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
4877  020252  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
4878                          ;SETUP REGISTER OUTPUT BUFFER FOR READ DATA COMMAND
4879  020256  012737  002037  001410          MOV     #2037,RMDA0
4880  020264  012737  001466  001436          MOV     #1466,RMDC0     ;CYL=822. TRK =4, SEC=31
4881  020272  012737  012000  001434          MOV     #FMT16!HCI,RMOF0        ;INHIBIT HEADER COMPARE
4882  020300  012702  054062                  MOV     #BUFONE,R2
4883  020304  010237  001406                  MOV     R2,RMBA0
4884  020310  012737  177400  001404          MOV     # C256.+1,RMWC0        ;256 WORDS
4885  020316  012737  000071  001402          MOV     #RD!GO,RMCS10           ;READ DATA COMMAND
4886  020324                          20$:
4887                          ;EXECUTE DATA COMMAND TO POINT WHERE SEARCH IS ENABLED USING SUBROUTINE
4888  020324  004737  023326                  JSR     PC,ENBSCH
4889  020330  000403                          BR      30$             ;GO TO 30$ IF NO ERROR
4890  020332  104000                          ERROR                   ;RETURN HERE IF ERROR
4891  020334  000137  021176                  JMP     190$
4892  020340                          30$:
4893                          ;FORCE SECTOR COMPARE USING SUBROUTINE
4894  020340  004737  024206                  JSR     PC,SCTCMP
4895  020344  000403                          BR      40$             ;GO TO 40$ IF NO ERROR
4896  020346  104000                          ERROR                   ;RETURN HERE IF ERROR
4897  020350  000137  021176                  JMP     190$
4898  020354                          40$:
4899                          ;SET "LOOKING FOR SYNC" USING SUBROUTINE
4900  020354  004737  024314                  JSR     PC,SETLFS
4901  020360  000403                          BR      50$
4902  020362  104000                          ERROR
4903  020364  000137  021176                  JMP     190$                    ``
4904  020370                          50$:
4905                          ;CLOCK THE SYNC PATTERN USING SUBROUTINE
4906  020370  004737  024476                  JSR     PC,CLKSNC
4907  020374  000403                          BR      55$
4908  020376  104000                          ERROR
4909  020400  000137  021176                  JMP     190$
4910                          ;;*****************************************************************
4911                          ;HEADER COMPARE IS INHIBITED FOR THIS TEST
4912                          ;"LOOKING FOR SYNC" SHOULD GO OFF WITHIN ONE CYCLE
4913  020404  012702  000002  55$:    MOV     #2,R2           ;ALLOW 2 PASSES THRU LOOP
4914  020410  000405                  BR      70$
4915  020412  004737  021310  60$:    JSR     PC,350$         ;RESET PROM STROBE
4916  020416  004737  021202          JSR     PC,300$         ;SET PROM STROBE
4917  020422  000240                  NOP
4918  020424                          70$:
4919
4920  020424  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4921  020430  032703  002000                  BIT     #PLFS,R3
4922  020434  001421                          BEQ     80$             ;BRANCH IF LOOKING FOR SYNC OFF
4923  020436  005302                          DEC     R2
4924  020440  001364                          BNE     60$
4925                          ;ERROR-LOOKING FOR SYNC DID NOT RESET DURING HEADER
4926  020442  010337  001142                  MOV     R3,$BDDAT               ;SETUP ERROR MESSAGE
4927  020446  042737  175777  001142          BIC     # CPLFS,$BDDAT
4928  020454  005037  001140                  CLR     $GDDAT
4929  020460  010037  001136                  MOV     R0,$BDADR
```

```
4930  020464  062737  000024  001136          ADD     #RMMR1,$BDADR
4931  020472  104040                          ERROR   40                      ;PLFS NOT RESET AGTER HEADER
4932  020474  000137  021176                  JMP     190$
4933  020500                          80$:
4934                                  ;;**************************************************************
4935                                  ;LOOKING FOR SYNC SHOULD COME ON FOR DATA AREA WITHIN 9 PROM CYCLES
4936  020500  012702  000011                  MOV     #9.,R2
4937  020504  004737  021310          85$:    JSR     PC,350$         ;SET STROBE OFF
4938  020510  004737  021202                  JSR     PC,300$         ;SET PROM STROBE
4939
4940  020514  016003  000024                  MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
4941  020520  032703  002000                  BIT     #PLFS,R3        ;
4942  020524  001022                          BNE     90$             ;BRANCH IF SYNC ENABLED
4943  020526  005302                          DEC     R2
4944  020530  001365                          BNE     85$
4945                                  ;ERROR-CAN'T LOOKING FOR SYNC DURING DATA AREA
4946  020532  012737  002000  001140          MOV     #PLFS,$GDDAT
4947  020540  010337  001142                  MOV     R3,$BDDAT
4948
4949  020544  042737  175777  001142          BIC     # CPLFS,$BDDAT
4950  020552  010037  001136                  MOV     R0,$BDADR
4951  020556  062737  000024  001136          ADD     #RMMR1,$BDADR
4952  020564  104035                          ERROR   35                      ;CAN'T SET PLFS DURING DATA
4953  020566  000137  021176                  JMP     190$
4954  020572                          90$:
4955  020572  004737  021310                  JSR     PC,350$         ;RESET PROM STROBE
4956                                  ;CLOCK THE SYNC PATTERN USING SUBROUTINE
4957  020576  004737  024476                  JSR     PC,CLKSNC
4958  020602  000402                          BR      100$
4959  020604  104000                          ERROR
4960  020606  000573                          BR      190$
4961  020610                          100$:
4962                                  ;;**************************************************************
4963                                  ;SIMULATE READ DATA
4964  020610  012702  000400                  MOV     #256.,R2        ;WORD COUNT
4965  020614  012703  000020          110$:   MOV     #16.,R3         ;BIT COUNT
4966  020620                          115$:
4967
4968  020620  012760  057401  000024          MOV     #MR1AAA!MRD!MCLK,RMMR1(R0)      ;LOAD RMMR1
4969
4970  020626  012760  053401  000024          MOV     #MR1AAA!MRD,RMMR1(R0)   ;LOAD RMMR1
4971  020634  005303                          DEC     R3
4972  020636  001370                          BNE     115$
4973  020640  005302                          DEC     R2
4974  020642  001364                          BNE     110$
4975                                  ;;**************************************************************
4976                                  ;SIMULATE ECC PATTERN
4977  020644  012737  177446  001174          MOV     #177446,$TMP0           ;FIRST ECC WORD
4978  020652  012737  015457  001176          MOV     #015457,$TMP1           ;SECOND ECC WORD
4979  020660  012702  001174                  MOV     #$TMP0,R2
4980  020664  012703  000020          120$:   MOV     #16.,R3
4981  020670  012704  055401          125$:   MOV     #MR1AAA!MCLK,R4
4982  020674  000241                          CLC
4983  020676  006012                          ROR     (R2)
4984  020700  103002                          BCC     130$
4985  020702  052704  002000                  BIS     #MRD,R4
```

```
4986  020706                          130$:
4987
4988  020706  010460  000024                MOV     R4,RMMR1(R0)    ;LOAD RMMR1
4989  020712  042704  004000                BIC     #MCLK,R4
4990
4991  020716  010460  000024                MOV     R4,RMMR1(R0)    ;LOAD RMMR1
4992  020722  005303                        DEC     R3
4993  020724  001361                        BNE     125$            ;CLOCK  A WORD
4994  020726  062702  000002                ADD     #2,R2
4995  020732  022702  001176                CMP     #$TMP1,R2
4996  020736  001752                        BEQ     120$            ;CLOCK THE NEXT WORD
4997                                  ;VERIFY DATA AREA AND READ GATE RESET
4998  020740  012702  000005                MOV     #5,R2
4999
5000  020744  004737  021202          140$: JSR     PC,300$         ;SET PROM STROBE
5001  020750  004737  021310                JSR     PC,350$         ;CLEAR PROM STROBE
5002
5003  020754  016003  000024                MOV     RMMR1(R0),R3    ;STORE RMMR1 AT R3
5004  020760  032703  000400                BIT     #PDA,R3
5005  020764  001417                        BEQ     150$
5006  020766  005302                        DEC     R2
5007  020770  001365                        BNE     140$
5008  020772  042703  177377                BIC     # CPDA,R3       ;SETUP ERROR MESSAGE
5009  020776  010337  001142                MOV     R3,$BDDAT
5010  021002  005037  001140                CLR     $GDDAT
5011  021006  010037  001136                MOV     R0,$BDADR
5012  021012  062737  000024  001136        ADD     #RMMR1,$BDADR
5013  021020  104063                        ERROR   63                      ;DATA AREA WON'T RESET
5014  021022  000465                        BR      190$
5015  021024                          150$:
5016
5017  021024  016003  000040                MOV     RMMR2(R0),R3    ;STORE RMMR2 AT R3
5018  021030  042703  176000                BIC     # C1777,R3
5019  021034  001413                        BEQ     160$
5020  021036  010337  001142                MOV     R3,$BDDAT
5021  021042  005037  001140                CLR     $GDDAT
5022  021046  010037  001136                MOV     R0,$BDADR
5023  021052  062737  000040  001136        ADD     #RMMR2,$BDADR
5024  021060  104053                        ERROR   53                      ;READ GATE WON'T RESET
5025  021062  000445                        BR      190$
5026  021064                          160$:
5027                                  ;;*****************************************************************
5028                                  ;VERIFY THERE ARE NO  ECC ERRORS
5029
5030  021064  016003  000014                MOV     RMER1(R0),R3    ;STORE RMER1 AT R3
5031  021070  042703  077677                BIC     # C<DCK!ECH>,R3
5032  021074  012737  000000  001140        MOV     #000000,$GDDAT
5033  021102  023703  001140                CMP     $GDDAT,R3
5034  021106  001411                        BEQ     170$
5035  021110  010337  001142                MOV     R3,$BDDAT
5036  021114  010037  001136                MOV     R0,$BDADR
5037  021120  062737  000014  001136        ADD     #RMER1,$BDADR
5038  021126  104064                        ERROR   64
5039  021130  000422                        BR      190$
5040  021132                          170$:
5041                                  ;;*****************************************************************
```

```
5042                                    ;VERIFY DATA BUFFER
5043  021132  012702  054062            MOV     #.UFONE,R2      ;DATA WAS STORED HERE
5044  021136  012703  177777            MOV     #177777,R3      ;EACH WORD ALL ONES
5045  021142  012704  000400            MOV     #256.,R4
5046  021146  022203              180$:  CMP     (R2)+,R3
5047  021150  001003                     BNE     185$
5048  021152  005304                     DEC     R4
5049  021154  001374                     BNE     180$
5050  021156  000407                     BR      190$            ;EXIT IF ALL THE LOCATIONS CHECKED
5051  021160  014237  001142      185$:  MOV     -(R2),$BDDAT    ;SETUP ERROR MESSAGE
5052  021164  010337  001140            MOV     R3,$GDDAT
5053  021170  010237  001136            MOV     R2,$BDADR
5054  021174  104065                     ERROR   65                      ;DATA INCORRECT DURING READ
5055  021176  000137  021414      190$:  JMP     400$            ;GREAT ESCAPE
5056  021202  012737  000021  021304  300$:  MOV     #17.,320$
5057                                    ;CLOCK PROM STROBE ON
5058  021210                    305$:
5059
5060  021210  012760  055401  000024        MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5061
5062  021216  012760  051401  000024        MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5063
5064  021224  016037  000024  021306        MOV     RMMR1(R0),330$  ;STORE RMMR1 AT 330$
5065  021232  042737  177737  021306        BIC     # CWC,330$
5066  021240  001020                     BNE     310$            ;EXIT IF PROM STROBE ON
5067  021242  005337  021304             DEC     320$
5068  021246  001360                     BNE     305$
5069  021250  013737  021304  001142        MOV     320$,$BDDAT     ;SETUP ERROR MESSAGE
5070  021256  012737  000040  001140        MOV     #WC,$GDDAT
5071  021264  010037  001136              MOV     R0,$BDADR
5072  021270  062737  000024  001136        ADD     #RMMR1,$BDADR
5073  021276  104030                      ERROR   30
5074  021300  000445                      BR      400$
5075  021302  000207              310$:   RTS     PC
5076  021304  000000              320$:   .WORD   0
5077
5078  021306  000000              330$:   .WORD   0
5079
5080  021310  012737  000021  021410  350$:  MOV     #17.,370$
5081                                    ;CLOCK THE STROBE OFF
5082  021316                    355$:
5083
5084  021316  012760  055401  000024        MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5085
5086  021324  012760  051401  000024        MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5087
5088  021332  016037  000024  021412        MOV     RMMR1(R0),380$  ;STORE RMMR1 AT 380$
5089  021340  042737  177737  021412        BIC     # CWC,380$
5090  021346  001417                      BEQ     360$
5091  021350  005337  021410              DEC     370$
5092  021354  001360                      BNE     355$
5093  021356  013737  021412  001142        MOV     380$,$BDDAT
5094  021364  005037  001140              CLR     $GDDAT
5095  021370  010037  001136              MOV     R0,$BDADR
5096  021374  062737  000024  001136        ADD     #RMMR1,$BDADR
5097  021402  104062                      ERROR   62              ;CAN'T CLEAR PROM STROBE
```

```
 5098  021404  000403                    BR      400$
 5099  021406  000207            360$:   RTS     PC
 5100  021410  000000            370$:   .WORD   0
 5101  021412  000000            380$:   .WORD   0
 5102  021414  000240            400$:   NOP             ;SUB-TEST EXIT POINT
```

```
5103  021416                          $EOSP:
5104  021416  000240                          NOP
5105  021420  013700  001456                  MOV     TSTQUE,R0
5106  021424  062700  000002                  ADD     #2,R0
5107  021430  010037  001456                  MOV     R0,TSTQUE
5108  021434  005710                          TST     (R0)
5109  021436  001402                          BEQ     1$
5110  021440  000137  004414                  JMP     READY
5111  021444  012737  001460  001456  1$:     MOV     #TSTQUE+2,TSTQUE
5112                                  .SBTTL  END OF PASS ROUTINE
5113
5114                                  ;;**********************************************************************
5115                                  ;*INCREMENT THE PASS NUMBER ($PASS)
5116                                  ;*TYPE "END PASS #XXXXX TOTAL NUMBER OF ERRORS SINCE LAST REPORT YYYYY"
5117                                  ;*WHERE XXXXX AND YYYYY ARE DECIMAL NUMBERS
5118                                  ;*IF THERES A MONITOR GO TO IT
5119                                  ;*IF THERE ISN'T JUMP TO READY
5120
5121  021452                          $EOP:
5122  021452  000240                          NOP
5123  021454  005037  001116                  CLR     $TSTNM          ;;ZERO THE TEST NUMBER
5124  021460  005037  001206                  CLR     $TIMES          ;;ZERO THE NUMBER OF ITERATIONS
5125  021464  005237  001230                  INC     $PASS           ;;INCREMENT THE PASS NUMBER
5126  021470  042737  100000  001230          BIC     #100000,$PASS   ;;DON'T ALLOW A NEG. NUMBER
5127  021476  005327                          DEC     (PC)+           ;;LOOP?
5128  021500  000001          $EOPCT: .WORD   1
5129  021502  003063                          BGT     $DOAGN          ;;YES
5130  021504  012737                          MOV     (PC)+,@(PC)+    ;;RESTORE COUNTER
5131  021506  000001          $ENDCT: .WORD   1
5132  021510  021500                          $EOPCT
5133  021512  104401  021520                  TYPE    ,65$            ;;TYPE ASCIZ STRING
5134  021516  000407                          BR      64$             ;;GET OVER THE ASCIZ
5135                          ;;65$:  .ASCIZ  <12><15>/END PASS #/
5136  021536                  64$:
5137  021536  013746  001230                  MOV     $PASS,-(SP)     ;;SAVE $PASS FOR TYPEOUT
5138                                                                  ;;TYPE PASS NUMBER
5139  021542  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
5140  021544  104401  021552                  TYPE    ,67$            ;;TYPE ASCIZ STRING
5141  021550  000421                          BR      66$             ;;GET OVER THE ASCIZ
5142                          ;;67$:  .ASCIZ  /  TOTAL ERRORS SINCE LAST REPORT /
5143  021614                  66$:
5144  021614  013746  001126                  MOV     $ERTTL,-(SP)    ;;SAVE $ERTTL FOR TYPEOUT
5145                                                                  ;;TOTAL NUMBER OF ERRORS
5146  021620  104405                          TYPDS                   ;;GO TYPE--DECIMAL ASCII WITH SIGN
5147  021622  104401  001217                  TYPE    ,$CRLF          ;;TYPE CARRIAGE RETURN, LINE FEED
5148  021626  005037  001126                  CLR     $ERTTL          ;;CLEAR ERROR TOTAL
5149  021632  013700  000042  $GET42: MOV     @#42,R0         ;;GET MONITOR ADDRESS
5150  021636  001405                          BEQ     $DOAGN          ;;BRANCH IF NO MONITOR
5151  021640  000005                          RESET                   ;;CLEAR THE WORLD
5152  021642  004710          $ENDAD: JSR     PC,(R0)         ;;GO TO MONITOR
5153  021644  000240                          NOP                     ;;SAVE ROOM
5154  021646  000240                          NOP                     ;;FOR
5155  021650  000240                          NOP                     ;;ACT11
5156  021652                  $DOAGN:
5157  021652  000137                          JMP     @(PC)+          ;;RETURN
5158  021654  004414          $RTNAD: .WORD   READY
```

```
 5159  021656      377      377      000  $ENULL: .BYTE  -1,-1,0          ;;NULL CHARACTER STRING
 5160           021662                            .EVEN
```

```
                                .SBTTL  SUBROUTINES

                                ;;*****************************************************************
                                .SBTTL  ERROR TYPEOUT ROUTINE

                                ;*THE ERROR TYPEOUT ROUTINE ASSEMBLES AND PRINTS INFORMATION
                                ;*REGARDING THE DETECTION OF AN ERROR AS FOLLOWS:
                                ;*
                                ;*      .UNIT NUMBER, TEST NUMBER, ERROR NUMBER AND PROGRAM COUNTER ARE
                                ;*PRINTED ON THE FIRST LINE;
                                ;*      .ERROR MESSAGE IS ASSEMBLED, FORMATTED AND PRINTED ON
                                ;*ONE OR MORE SUCCEEDING LINES;
                                ;*      .PAIRED LINES OF ERROR HEADERS AND ERROR DATA
                                ;*ARE PRINTED AFTER THE ERROR MESSAGE.

021662                          ERRTYP:
021662  104414                          SAVREG
021664  032777  020000  157262          BIT     #SW13,@SWR      ;INHIBIT TYPEOUTS??
021672  001402                          BEQ     1$              ;NO!!
021674  000137  022412                  JMP     21$             ;YES!!
                                ;TYPE UNIT NUMBER, TEST NUMBER, ERROR NUMBER, AND PROGRAM COUNTER
021700  104401  001217          1$:     TYPE    ,$CRLF
021704  104401  022426                  TYPE    ,ERTY00         ;TYPE "UNT#"
021710  013746  001234                  MOV     $UNIT,-(SP)     ;;SAVE $UNIT FOR TYPEOUT
                                                                ;;TYPE UNIT NUMBER
021714  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
021716     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
021717     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
021720  005037  022416                  CLR     TSTNMB          ;LOAD TEST NUMBER FOR
021724  013737  001226  022416          MOV     $TESTN,TSTNMB
021732  104401  022434                  TYPE    ,ERTY01         ;TYPE "TST#"
021736  013746  022416                  MOV     TSTNMB,-(SP)    ;;SAVE TSTNMB FOR TYPEOUT
                                                                ;;TYPE TEST NUMBER
021742  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
021744     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
021745     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
021746  005037  022420                  CLR     ERRNMB          ;LOAD ERROR NUMBER FOR
021752  113737  001130  022420          MOVB    $ITEMB,ERRNMB   ;TYPEOUT
021760  001406                          BEQ     2$              ;SKIP IF NO ERROR CALLED
021762  104401  022444                  TYPE    ,ERTY02         ;TYPE "ERR#"
021766  013746  022420                  MOV     ERRNMB,-(SP)    ;;SAVE ERRNMB FOR TYPEOUT
                                                                ;;TYPE ERROR NUMBER
021772  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
021774     003                          .BYTE   3               ;;TYPE 3 DIGIT(S)
021775     000                          .BYTE   0               ;;SUPPRESS LEADING ZEROS
021776  104401  022453          2$:     TYPE    ,ERTY03         ;TYPE "PC="
022002  013746  001132                  MOV     $ERRPC,-(SP)    ;;SAVE $ERRPC FOR TYPEOUT
                                                                ;;TYPE PROGRAM COUNTER
022006  104403                          TYPOS                   ;;GO TYPE--OCTAL ASCII
022010     006                          .BYTE   6               ;;TYPE 6 DIGIT(S)
022011     001                          .BYTE   1               ;;TYPE LEADING ZEROS
                                ;GENERATE POINTER TO ERROR TABLE UNLESS ERROR NUMBER IS 0
022012  005737  022420          3$:     TST     ERRNMB          ;WAS AN ERROR CALLED?
022016  001575                          BEQ     21$             ;NO!!
022020  104401  001217                  TYPE    ,$CRLF          ;YES-TYPE CRLF
022024  105037  022424                  CLRB    BOTFLG          ;CLEAR BOT FLAG
```

```
5217  022030  105037  022425              CLRB    CHRCNT          ;CLEAR CHARACTER COUNTER
5218  022034  013700  022420              MOV     ERRNMB,R0       ;R0 POINTS TO FIRST OF
5219  022040  006300                      ASL     R0              ;FOUR ENTRIES IN ERROR
5220  022042  006300                      ASL     R0              ;TABLE
5221  022044  006300                      ASL     R0
5222  022046  062700  001522              ADD     #$ERRTB-8.,R0
5223  022052  011001                      MOV     (R0),R1         ;R1 POINTS TO ERROR MESSAGE
5224                                                              ;TABLE
5225  022054  001507                      BEQ     12$             ;BRANCH IF NO ERROR MESSAGE
5226                                ;TYPE THE ERROR MESSAGE
5227  022056  012102                4$:   MOV     (R1)+,R2        ;R2=ADDRESS OF MESSAGE STRING
5228  022060  001505                      BEQ     12$             ;BRANCH IF END OF MESSAGE
5229  022062  010237  022230              MOV     R2,11$          ;LOAD ADDRESS OF STRING
5230  022066  005037  022422              CLR     BOTADR          ;CLEAR BOT ADDRESS
5231  022072  112203                5$:   MOVB    (R2)+,R3        ;END OF STRING??
5232  022074  001454                      BEQ     10$             ;YES!!
5233  022076  122703  000015              CMPB    #CR,R3          ;CARRIAGE RETURN??
5234  022102  001003                      BNE     6$              ;NO!!
5235  022104  105037  022425              CLRB    CHRCNT          ;YES-CLEAR CHAR COUNT
5236  022110  000770                      BR      5$              ;GET NEXT CHARACTER
5237  022112  122703  000012        6$:   CMPB    #LF,R3          ;LINE FEED??
5238  022116  001765                      BEQ     5$              ;YES-GET NEXT CHARACTER
5239  022120  122703  000011              CMPB    #HT,R3          ;HORIZONTAL TAB??
5240  022124  001007                      BNE     8$              ;NO!!
5241  022126  105237  022425        7$:   INCB    CHRCNT          ;ADJUST CHARACTER COUNT
5242  022132  132737  000007  022425      BITB    #7,CHRCNT
5243  022140  001372                      BNE     7$
5244  022142  000407                      BR      9$
5245  022144  105237  022425        8$:   INCB    CHRCNT          ;INCREMENT CHARACTER COUNT
5246  022150  122703  000040              CMPB    #' ,R3          ;SPACE??
5247  022154  001002                      BNE     9$              ;NO!!
5248  022156  010237  022422              MOV     R2,BOTADR       ;SAVE ADDRESS OF SPACE
5249  022162  122737  000100  022425  9$: CMPB    #64.,CHRCNT     ;END OF LINE??
5250  022170  103340                      BHIS    5$              ;NO!!
5251  022172  013704  022422              MOV     BOTADR,R4       ;GET ADDRESS OF LAST SPACE
5252  022176  001007                      BNE     90$             ;BRANCH IF SPACE DETECTED
5253  022200  104401  001217              TYPE    ,$CRLF          ;TYPE CRLF
5254  022204  105037  022425              CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
5255  022210  013702  022230              MOV     11$,R2          ;SET UP R2 FOR TESTING
5256  022214  000726                      BR      5$
5257  022216  105044                90$:  CLRB    -(R4)           ;REPLACE SPACE
5258  022220  112737  177777  022424      MOVB    #-1,BOTFLG      ;SET BOT FLAG
5259  022226  104401                10$:  TYPE                    ;TYPE ERROR MESSAGE STRING
5260  022230  000000                11$:  .WORD                   ;STRING ADDRESS GOES HERE
5261  022232  105737  022424              TSTB    BOTFLG          ;WAS STRING TRUNCATED??
5262  022236  001707                      BEQ     4$              ;NO!!
5263  022240  104401  001217              TYPE    ,$CRLF          ;YES-TYPE CRLF
5264  022244  105037  022424              CLRB    BOTFLG          ;CLEAR BOT FLAG
5265  022250  105037  022425              CLRB    CHRCNT          ;CLEAR CHARACTER COUNT
5266  022254  013702  022422              MOV     BOTADR,R2       ;SETUP R2 FOR TESTING
5267  022260  010237  022230              MOV     R2,11$          ;SETUP 11$ FOR TYPING
5268  022264  112742  000040              MOVB    #' ,-(R2)       ;RESTORE SPACE
5269  022270  105722                      TSTB    (R2)+           ;RESTORE R2
5270  022272  000677                      BR      5$              ;TYPE REST OF STRING
5271  022274                        12$:
5272                                ;TYPE ERROR HEADER AND ERROR DATA
```

```
5273  022274                          13$:
5274  022274  016001  000002                  MOV     2(R0),R1        ;R1 POINTS TO ERROR HEADER TABLE
5275  022300  001444                          BEQ     21$             ;BRANCH IF NO HEADER
5276  022302  104401  001217                  TYPE    ,$CRLF          ;(ASSUME NO DATA)
5277  022306  016002  000004                  MOV     4(R0),R2        ;R2 POINTS TO DATA ADDRESS TABLE
5278  022312  016003  000006                  MOV     6(R0),R3        ;R3 POINTS TO FORMAT TABLE
5279  022316  012137  022326          14$:    MOV     (R1)+,15$       ;PUT HEADER ADDRESS FOR TYPE
5280  022322  001433                          BEQ     21$             ;BRANCH IF END OF HEADERS
5281                                                                  ;(ASSUME END OF DATA)
5282  022324  104401                          TYPE
5283  022326  000000                  15$:    .WORD   0               ;HEADER ADDRESS GOES HERE
5284  022330  104401  001217                  TYPE    ,$CRLF
5285  022334  005702                          TST     R2              ;DATA WITH HEADER??
5286  022336  001767                          BEQ     14$             ;NO!!
5287  022340  012204                          MOV     (R2)+,R4        ;R4 POINTS TO DATA ADDRESS
5288  022342  012305                          MOV     (R3)+,R5        ;R5 POINTS TO FORMAT
5289  022344  105725                  16$:    TSTB    (R5)+           ;WHAT KIND OF DATA??
5290  022346  100407                          BMI     18$             ;BINARY
5291  022350  001403                          BEQ     17$             ;OCTAL
5292  022352  013446                          MOV     @(R4)+,-(SP)    ;DECIMAL
5293  022354  104405                          TYPDS
5294  022356  000405                          BR      19$
5295  022360  013446                  17$:    MOV     @(R4)+,-(SP)
5296  022362  104402                          TYPOC
5297  022364  000402                          BR      19$
5298  022366  013446                  18$:    MOV     @(R4)+,-(SP)
5299  022370  104406                          TYPBN
5300  022372  005714                  19$:    TST     (R4)            ;MORE DATA??
5301  022374  001403                          BEQ     20$             ;NO!!
5302  022376  104401  022461                  TYPE    ,ERTY04         ;YES-TYPE 2 SPACES
5303  022402  000760                          BR      16$             ;AND CONTINUE
5304  022404  104401  001217          20$:    TYPE    ,$CRLF          ;TYPE ONE BLANK LINE
5305  022410  000742                          BR      14$             ;BEFORE NEXT HEADER
5306  022412  104415                  21$:    RESREG
5307  022414  000207                          RTS     PC
5308
5309  022416  000000                  TSTNMB: .WORD   0               ;TEST NUMBER
5310  022420  000000                  ERRNMB: .WORD   0               ;ERROR NUMBER
5311  022422  000000                  BOTADR: .WORD   0               ;BEGINNING OF TEXT ADDRESS
5312  022424     000                  BOTFLG: .BYTE   0               ;BOT FLAG
5313  022425     000                  CHRCNT: .BYTE   0               ;CHARACTER COUNT
5314
5315  022426  047125  052111  000043  ERTY00: .ASCIZ  @UNIT#@
5316  022434  020054  042524  052123  ERTY01: .ASCIZ  @, TEST#@
5317  022442  000043
5318  022444  020054  051105  021522  ERTY02: .ASCIZ  @, ERR#@
5319  022452     000
5320  022453     054  050040  036503  ERTY03: .ASCIZ  @, PC=@
5321  022460     000
5322  022461     040  000040          ERTY04: .ASCIZ  @  @
5323                                  .EVEN
5324
```

```
                                .SBTTL  CLOCK SUBROUTINES

                                ;ROUTINE TO SIZE FOR CLOCKS (KW11-L OR KW11-P)
                                ;;*****************************************************************
022464  000240                  SIZCLK: NOP
022466  013746  000004                  MOV     ERRVEC,-(SP)    ;;PUSH ERRVEC ON STACK
022472  013746  000006                  MOV     ERRVEC+2,-(SP)          ;;PUSH ERRVEC+2 ON STACK
022476  012737  022562  000004          MOV     #10$,ERRVEC     ;LOAD 04 TRAP VECTORS
022504  012737  000300  000006          MOV     #PR6,ERRVEC+2

                                ;SEE IF A KW11-P CLOCK IS PRESENT - GO TO 10$ IF NOT PRESENT
022512  005777  156764                  TST     @$LPCSR         ;TEST FOR P CLOCK
022516  012737  022724  001526          MOV     #PCLOCK,CLOCK   ;LOAD SUBROUTINE ADDRESS
022524  012737  023046  001530          MOV     #PSTOP,STOP     ;LOAD STOP ADDRESS
022532  012777  023012  156746          MOV     #PCOUNT,@$LPVEC ;LOAD P CLOCK INTERRUPT VECTOR
022540  012777  000300  156742          MOV     #PR6,@$LPVEC+2
022546  013777  001516  156740          MOV     $LLVEC+2,@$LLVEC;CLEAR L CLOCK INTERRUPT VECTOR
022554  005077  156736                  CLR     @$LLVEC+2
022560  000454                          BR      30$
022562  012716  022570          10$:    MOV     #15$,(SP)       ;DUMMY RTI ADDRESS
022566  000002                          RTI                     ;RESTORE PRIORITY
022570                          15$:
                                ;NO P CLOCK-SEE IF L CLOCK IS PRESENT-GO TO 20$ IF NOT PRESET
022570  012737  022646  000004          MOV     #20$,ERRVEC     ;CHANGE 04 TRAP VECTOR
022576  005777  156710                  TST     @$LLCSR         ;TEST FOR L CLOCK
022602  012737  022742  001526          MOV     #LCLOCK,CLOCK   ;LOAD SUBROUTINE ADDRESS
022610  012737  023054  001530          MOV     #LSTOP,STOP     ;LOAD STOP ADDRESS
022616  012777  023012  156670          MOV     #LCOUNT,@$LLVEC ;LOAD L CLOCK INTERRUPT VECTOR
022624  012777  000300  156664          MOV     #PR6,@$LLVEC+2
022632  013777  001510  156646          MOV     $LPVEC+2,@$LPVEC;CLEAR P CLOCK INTERRUPT VECTOR
022640  005077  156644                  CLR     @$LPVEC+2
022644  000422                          BR      30$
022646  012716  022654          20$:    MOV     #25$,(SP)       ;DUMMY RTI ADDRESS
022652  000002                          RTI                     ;RESTORE PRIORITY
022654                          25$:
                                ;NO CLOCK AVAILABLE - AUGMENT RETURN ADDRESS
022654  005037  001526                  CLR     CLOCK           ;CLEAR SUBROUTINE ADDRESS
022660  012737  001510  001506          MOV     #$LPVEC+2,$LPVEC;CLEAR P CLOCK INTERRUPT VECTOR
022666  005037  001510                  CLR     $LPVEC+2
022672  012737  001516  001514          MOV     #$LLVEC+2,$LLVEC;CLEAR L CLOCK INTERRUPT VECTOR
022700  005037  001516                  CLR     $LLVEC+2
022704  062766  000002  000004          ADD     #2,4(SP)        ;CHANGE RETURN ADDRESS

022712                          30$:
022712  012637  000006                  MOV     (SP)+,ERRVEC+2          ;;POP STACK INTO ERRVEC+2
022716  012637  000004                  MOV     (SP)+,ERRVEC    ;;POP STACK INTO ERRVEC
022722  000207                          RTS     PC
                                ;ROUTINES TO START THE CLOCK (KW11-L OR KW11-P)
                                ;;*****************************************************************
022724  012777  177777  156552  PCLOCK: MOV     #-1,@$LPCSB     ;LOAD COUNT SET BUFFER
022732  012777  000135  156542          MOV     #135,@$LPCSR    ;LOAD CONTROL REGISTER
022740  000403                          BR      PLCLK           ;GO TO COMMON CODE

022742  012777  000100  156542  LCLOCK: MOV     #100,@$LLCSR    ;LOAD CONTROL REGISTER
```

```
5381  022750  005037  001522          PLCLK:  CLR     TIME            ;CLEAR TIMER COUNT
5382  022754  104400                          TRAP                    ;;PUSH OLD PSW AND PC ON STACK
5383  022756  012605                          MOV     (SP)+,R5        ;;SAVE THE PSW IN R5
5384  022760  010537  001520                  MOV     R5,$PSW         ;SAVE PRIORITY
5385  022764  042705  177437                  BIC     # CPR7,R5       ;MASK X
5386  022770  022705  000300                  CMP     #PR6,R5         ;IS PRIORITY TOO HIGH??
5387  022774  101005                          BHI     40$             ;NO!!
5388  022776  012746  000240                  MOV     #PR5,-(SP)      ;;PUT NEW PS ON STACK
5389  023002  012746  023010                  MOV     #30$,-(SP)      ;;PUT NEW PC ON  STACK
5390  023006  000002                          RTI                     ;;POP NEW PC AND PS
5391  023010                          30$:
5392  023010  000207                  40$:    RTS     PC
5393
5394                                  ;ROUTINES TO HANDLE CLOCK INTERRUPTS (KW11-L OR KW11-P)
5395
5396  023012                          PCOUNT:
5397  023012                          LCOUNT:
5398  023012  062737  000021  001522          ADD     #17.,TIME       ;ADD 17MS TO ELAPSED TIME
5399  023020  103003                          BCC     10$             ;BRANCH IF NO OVERFLOW
5400  023022  012737  177777  001522          MOV     #-1,TIME        ;RESTORE MAXIMUM COUNT
5401  023030  162737  000021  001524  10$:    SUB     #17.,WATCH      ;DECREMENT REMAINING TIME
5402  023036  100002                          BPL     20$             ;BRANCH IF POSITIVE
5403  023040  005037  001524                  CLR     WATCH           ;CLEAR REMAINING TIME
5404  023044  000002                  20$:    RTI                     ;RETURN TO USER
5405
5406                                  ;ROUTINES TO STOP THE CLOCK (KW11-L OR KW11-P)
5407
5408  023046  005077  156430          PSTOP:  CLR     @$LPCSR         ;STOP P CLOCK
5409  023052  000402                          BR      PLSTP           ;GO TO COMMON STOP CODE
5410
5411  023054  005077  156432          LSTOP:  CLR     @$LLCSR         ;STOP L CLOCK
5412
5413  023060                          PLSTP:
5414  023060  013746  001520                  MOV     $PSW,-(SP)      ;;PUT NEW PS ON STACK
5415  023064  012746  023072                  MOV     #10$,-(SP)      ;;PUT NEW PC ON  STACK
5416  023070  000002                          RTI                     ;;POP NEW PC AND PS
5417  023072                          10$:
5418  023072  000207                          RTS     PC
5419
5420                                  .SBTTL  SET VOLUME VALID SUBROUTINE
5421
5422                                  ;THIS SUBROUTINE INITIALIZES THE SUBSYSTEM AND SETS VOLUME VALID,
5423                                  ;RETURNING WITH THE DRIVE STILL IN DIAGNOSTIC MODE.  THE SUBROUTINE
5424                                  ;RETURNS TO THE WORD FOLLOWING THE CALL, EXCEPT WHEN AN ERROR IS
5425                                  ;DETECTED, IN WHICH CASE IT RETURNS TO THE SECOND WORD FOLLOWING THE
5426                                  ;CALL.
5427                                  ;
5428                                  ;CALL:  JSR     PC,SETVV        JUMP TO SUBROUTINE
5429                                  ;       BR      ??              RETURN HERE IF NO ERROR
5430                                  ;       ERROR                   RETURN HERE IF ERROR
5431
5432  023074                          SETVV:
5433  023074  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
5434  023102  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
5435
5436  023106  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
```

```
5437
5438  023114  012760  001001  000024          MOV     #DMD.MUR,RMMR1(R0)      ;LOAD RMMR1
5439
5440  023122  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5441
5442  023130  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5443
5444  023136  012760  000023  000000          MOV     #PACACK!GO,RMCS1(R0)    ;LOAD RMCS1
5445
5446  023144  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
5447  023152  042737  177677  001142          BIC     # CVV,$BDDAT
5448  023160  001020                          BNE     10$             ;BRANCH IF VOLUME VALID SET
5449  023162  010037  001136                  MOV     R0,$BDADR       ;SETUP FOR ERROR MSG
5450  023166  062737  000012  001136          ADD     #RMDS,$BDADR
5451  023174  012737  000100  001140          MOV     #VV,$GDDAT
5452  023202  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN ADDRESS TO ERROR
5453  023206  112776  000100  000000          MOVB    #100,@(SP)      ;WRITE ERROR NUMBER
5454  023214  012737  000022  001174          MOV     #PACACK,$TMP0
5455  023222  000207                  10$:    RTS     PC              ;RETURN
5456
5457                                  .SBTTL  SET OFFSET MODE SUBROUTINE
5458
5459                                  ;THIS SUBROUTINE EXECUTES AN OFFSET COMMAND AND VERIFIES THAT OFFSET
5460                                  ;MODE SETS.  THE DRIVE SHOULD BE IN DIAGNOSTIC MODE WHEN CALLING THE
5461                                  ;SUBROUTINE, WHICH WILL LEAVE DMD ON.  THE SUBROUTINE RETURNS TO THE
5462                                  ;WORD FOLLOWING THE CALL UNLESS THERE IS AN ERROR, IN WHICH CASE IT
5463                                  ;RETURNS TO THE SECOND WORD FOLLOWING THE CALL
5464                                  ;
5465                                  ;CALL:  JSR     PC,SETOM        JUMP TO SUBROUTINE
5466                                  ;       BR      ??              RETURN HERE IF NO ERROR
5467                                  ;       ERROR                   RETURN HERE IF ERROR
5468
5469  023224                          SETOM:
5470
5471  023224  012760  001001  000024          MOV     #DMD'MUR,RMMR1(R0)      ;LOAD RMMR1
5472
5473  023232  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5474
5475  023240  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5476
5477  023246  012760  000015  000000          MOV     #OFFSET!GO,RMCS1(R0)    ;LOAD RMCS1
5478
5479  023254  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
5480  023262  042737  177776  001142          BIC     # COM,$BDDAT
5481  023270  001015                          BNE     10$             ;BRANCH IF OFFSET ON
5482  023272  012737  000001  001140          MOV     #OM,$GDDAT
5483  023300  010037  001136                  MOV     R0,$BDADR
5484  023304  062737  000012  001136          ADD     #RMDS,$BDADR
5485  023312  062716  000002                  ADD     #2,(SP)         ;MOVE RETURN ADDRESS TO ERROR
5486  023316  112776  000111  000000          MOVB    #111,@(SP)      ;WRITE ERROR NUMBER
5487  023324                          10$:
5488  023324  000207                          RTS     PC              ;RETURN TO USER
5489                                  .SBTTL  ENABLE SEARCH SUBROUTINE
5490                                  ;;*****************************************************************
5491
5492  023326                          ENBSCH:
```

```
 5493
 5494                                   ;THE REGISTER OUTPUT BUFFER  SHOULD CONTAIN:
 5495                                   ;       RMDA0   = DESIRED SECTOR AND TRACK ADDRESS
 5496                                   ;       RMDC0   = DESIRED CYLINDER ADDRESS
 5497                                   ;       RMOF0   = FORMAT, ECI, HCI
 5498                                   ;       RMBA0   = BUFFER ADDRESS
 5499                                   ;       RMWC0   = WORD COUNT
 5500                                   ;       RMCS10  = FUNCTION CODE
 5501
 5502                                   ;NOTE:  OFFSET IS NOT ENABLED WHEN USING THIS SUBROUTINE
 5503
 5504                                   ;;*****************************************************************
 5505                                   ;SET VOLUME VALID
 5506                                   ;FIRST,CLEAR -SET DIAGNOSTIC MODE-SYNCHRONIZE
 5507                                   ;PROM STROBE
 5508  023326  012760  000040  000010          MOV     #CLR,RMCS2(R0)  ;CLEAR THE MASSBUS
 5509  023334  111160  000010                  MOVB    (R1),RMCS2(R0)  ;SELECT UNIT
 5510
 5511  023340  012760  000001  000024          MOV     #DMD,RMMR1(R0)  ;LOAD RMMR1
 5512  023346  012704  000041                  MOV     #33.,R4         ;ALLOW UP TO 33 BIT CLOCLS
 5513                                                                  ;TO SYNC PROM STROBE
 5514                                   ; .....
 5515  023352                         5$:
 5516
 5517  023352  012760  005001  000024          MOV     #DMD!MUR!MCLK,RMMR1(R0) ;LOAD RMMR1
 5518
 5519  023360  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)      ;LOAD RMMR1
 5520
 5521  023366  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
 5522  023372  032705  000040                  BIT     #WC,R5  ;WAIT FOR PROM STROBE TO COME ON
 5523  023376  001023                          BNE     6$
 5524  023400  005304                          DEC     R4
 5525  023402  001363                          BNE     5$
 5526  023404  010037  001136                  MOV     R0,$BDADR       ;PROM STROBE WONT SET
 5527  023410  062737  000024  001136          ADD     #RMMR1,$BDADR
 5528  023416  005037  001142                  CLR     $BDDAT
 5529  023422  012737  000040  001140          MOV     #WC,$GDDAT
 5530  023430  062716  000002                  ADD     #2,(SP)
 5531  023434  112776  000030  000000          MOVB    #30,@(SP)
 5532  023442  000137  024204                  JMP     60$             ;REPROT ERROR
 5533  023446                         6$:
 5534
 5535  023446  012760  005001  000024          MOV     #DMD!MUR!MCLK,RMMR1(R0) ;LOAD RMMR1
 5536
 5537  023454  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)      ;LOAD RMMR1
 5538
 5539  023462  016005  000024                  MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
 5540  023466  032705  000040                  BIT     #WC,R5          ;WAIT FOR PROM STROBE TO GO OFF
 5541  023472  001423                          BEQ     7$
 5542  023474  005304                          DEC     R4
 5543  023476  001363                          BNE     6$
 5544  023500  010037  000136                  MOV     R0,$BDADR       ;PROM STROBE WONT RESET
 5545  023504  062737  000024  001136          ADD     #RMMR1,$BDADR   ;
 5546  023512  012737  000040  001142          MOV     #WC,$BDDAT
 5547  023520  005037  001140                  CLR     $GDDAT
 5548  023524  062716  000002                  ADD     #2,(SP)
```

```
5549  023530  112776  000062  000000          MOVB    #62,@(SP)
5550  023536  000137  024204                  JMP     60$
5551  023542                          7$:
5552                                  ;SECOND, CLOCK INDEX PULSE TO 1. CLEAR FORMAT CHANGE FLOP
5553                                  ;                               2. CLEAR RTC FLOP
5554
5555  023542  012760  001005  000024          MOV     #DMD:MUR!MI,RMMR1(R0)   ;LOAD RMMR1
5556
5557  023550  012760  001001  000024          MOV     #DMD!MUR,RMMR1(R0)      ;LOAD RMMR1
5558                                  ... ...
5559
5560  023556  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5561
5562  023564  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5563
5564  023572  012760  000023  000000          MOV     #PACACK:GO,RMCS1(R0)    ;LOAD RMCS1
5565
5566  023600  016037  000012  001142          MOV     RMDS(R0),$BDDAT ;STORE RMDS AT $BDDAT
5567  023606  042737  177677  001142          BIC     # CVV,$BDDAT    ;VOLUME VALID SHOULD BE SET
5568  023614  001021                          BNE     10$
5569  023616  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MSG
5570  023622  062737  000012  001136          ADD     #RMDS,$BDADR
5571  023630  012737  000100  001140          MOV     #VV,$GDDAT
5572  023636  062716  000002                  ADD     #2,(SP)         ;WRITE ERROR NUMBER
5573  023642  112776  000100  000000          MOVB    #100,@(SP)      ;
5574  023650  012737  000022  001174          MOV     #PACACK,$TMP0
5575  023656  000552                          BR      60$
5576  023660                          10$:
5577                                  ;;**************************************************************
5578                                  ;LOAD REGISTERS
5579
5580  023660  013760  001410  000006          MOV     RMDA0,RMDA(R0)  ;LOAD RMDA
5581
5582  023666  013760  001436  000034          MOV     RMDC0,RMDC(R0)  ;LOAD RMDC
5583
5584  023674  013760  001434  000032          MOV     RMOF0,RMOF(R0)  ;LOAD RMOF
5585
5586  023702  013760  001404  000002          MOV     RMWC0,RMWC(R0)  ;LOAD RMWC
5587
5588  023710  013760  001406  000004          MOV     RMBA0,RMBA(R0)  ;LOAD RMBA
5589                                  ;;**************************************************************
5590                                  ;ENABLE DEBUG CLOCK AND LOAD FUNCTION CODE
5591
5592  023716  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
5593
5594  023724  012760  000000  000014          MOV     #0,RMER1(R0)    ;LOAD RMER1
5595
5596  023732  012760  000000  000042          MOV     #0,RMER2(R0)    ;LOAD RMER2
5597
5598  023740  013760  001402  000000          MOV     RMCS10,RMCS1(R0)        ;LOAD RMCS1
5599                                  ;;**************************************************************
5600                                  ;WAIT FOR "RUN AND GO" TO SET
5601  023746  012737  000310  001524          MOV     #200.,WATCH     ;SET WATCHDOG TIMER VALUE
5602  023754  004777  155546                  JSR     PC,@CLOCK       ;START THE CLOCK
5603  023760                          20$:
5604
```

```
                        ENABLE SEARCH SUBROUTINE

5605  023760  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
5606  023766  042737  137777  001142          BIC     # CRG,$BDDAT
5607  023774  001023                          BNE     30$
5608  023776  005737  001524                  TST     WATCH           ;ANY TIME LEFT?
5609  024002  001366                          BNE     20$             ;YES-WAIT
5610  024004  004777  155520                  JSR     PC,@STOP        ;STOP THE CLOCK
5611  024010  012737  040000  001140          MOV     #RG,$GDDAT      ;SETUP ERROR MSG
5612  024016  010037  001136                  MOV     R0,$BDADR
5613  024022  062737  000024  001136          ADD     #RMMR1,$BDADR
5614  024030  062716  000002                  ADD     #2,(SP)         ;WRITE ERROR NUMBER IN
5615  024034  112776  000101  000000          MOVB    #101,@(SP)              ;USER'S CALL -
5616
5617  024042  000460                          BR      60$
5618  024044                          30$:
5619  024044  004777  155460                  JSR     PC,@STOP        ;STOP THE CLOCK
5620                          ;;*****************************************************************
5621                          ;STEP COMMAND SEQUENCER TO HEAD SEQUENCE AT LOCATION 156 (17 CLOCKS)
5622  024050  012704  000021                  MOV     #17.,R4         ;R4=CLOCK COUNT
5623  024054                          40$:
5624
5625  024054  012760  141401  000024          MOV     #DMD!MUR!DBEN!MOC!DBCK,RMMR1(R0)        ;LOAD RMMR1
5626
5627  024062  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
5628  024070  005304                          DEC     R4
5629  024072  001370                          BNE     40$
5630                          ;;*****************************************************************
5631                          ;DROP ON CYLINDER TO RESET LATCH, THEN SET ON CYLINDER TO PASS
5632                          ;TEST AT LOCATION 166
5633
5634  024074  012760  041001  000024          MOV     #DMD!MUR!DBEN,RMMR1(R0) ;LOAD RMMR1
5635
5636  024102  012760  041401  000024          MOV     #DMD!MUR!DBEN!MOC,RMMR1(R0)     ;LOAD RMMR1
5637                          ;;*****************************************************************
5638                          ;STEP SEQUENCER (35 CLOCKS) AND VERIFY SEARCH IS ENABLED
5639  024110  012704  000043                  MOV     #35.,R4         ;R4=CLOCK COUNT
5640  024114                          50$:
5641
5642  024114  012760  151401  000024          MOV     #DMD!MUR!DBEN.MOC!DBCK!DTO,RMMR1(R0)    ;LOAD RMMR1
5643
5644  024122  012760  051401  000024          MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5645  024130  005304                          DEC     R4
5646  024132  001370                          BNE     50$
5647
5648  024134  016037  000024  001142          MOV     RMMR1(R0),$BDDAT        ;STORE RMMR1 AT $BDDAT
5649  024142  042737  173777  001142          BIC     # CESRC,$BDDAT  ;SEARCH SHOULD BE ENABLED
5650  024150  001015                          BNE     60$
5651  024152  010037  001136                  MOV     R0,$BDADR       ;SETUP ERROR MESSAGE
5652  024156  062737  000024  001136          ADD     #RMMR1,$BDADR
5653  024164  012737  004000  001140          MOV     #ESRC,$GDDAT
5654  024172  062716  000002                  ADD     #2,(SP)         ;WRITE ERROR NUMBER
5655  024176  112776  000102  000000          MOVB    #102,@(SP)
5656  024204                          60$:
5657                          ;;*****************************************************************
5658                          ;RETURN TO USER
5659  024204  000207                          RTS     PC
```

```
5660                                    .SBTTL  SECTOR COMPARE SUBROUTINE
5661
5662                                    ;SECTOR COMPARE SUBROUTINE
5663
5664                                    ;THIS SUBROUTINE CONTINUES THE EXECUTION OF A DATA COMMAND
5665                                    ;FROM WHERE SEARCH HAS BEEN ENABLED TO WHERE SECTOR COMPARE
5666                                    ;IS SET.
5667
5668  024206                            SCTCMP:
5669
5670                                    ;;*****************************************************************
5671                                    ;SET INDEX PULSE TO CLEAR FORMAT CHANGE FLOP
5672
5673  024206  012760  051405  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MI,RMMR1(R0)      ;LOAD RMMR1
5674
5675  024214  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5676                                    ;WAIT AT LEAST 4 MS FOR "RETURN TO CENTER LINE" ONE SHOT TO SET
5677  024222  012737  000006  001524            MOV     #6,WATCH        ;SET WATCHDOG TIMER VALUE
5678  024230  004777  155272                    JSR     PC,@CLOCK       ;START THE CLOCK
5679  024234  005737  001524           10$:     TST     WATCH
5680  024240  001375                            BNE     10$
5681  024242  004777  155262                    JSR     PC,@STOP        ;STOP THE CLOCK
5682                                    ;DATA INPUT TO SEARCH ENABLE FLOP SHOULD BE ONE - CLOCK SECTOR PULSE TO
5683                                    ;SET FLOP
5684
5685  024246  012760  051441  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MS,RMMR1(R0)      ;LOAD RMMR1
5686
5687  024254  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5688                                    ;WITH SECTOR COMPARE HIGH, CLOCK SECTOR PULSE TO SET SECTOR COMPARE FLOP
5689
5690  024262  012760  051403  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
5691
5692  024270  012760  051443  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MSC!MS,RMMR1(R0)  ;LOAD RMMR1
5693
5694  024276  012760  051403  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MSC,RMMR1(R0)     ;LOAD RMMR1
5695
5696  024304  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5697
5698                                    ;;*****************************************************************
5699                                    ;RETURN TO USER
5700  024312  000207                            RTS     PC
5701
5702                                    ;;*****************************************************************
5703                                    .SBTTL  SET LOOKING FOR SYNC SUBROUTINE
5704                                    ;THIS SUBROUTINE WILL SETUP THE DATA TIMING SEQUENCER
5705                                    ;ASSUMING SEARCH IS ENABLED,TO THE POINT WHERE "PLFS" IS ACTIVE
5706
5707  024314                            SETLFS:
5708                                    ;PULSE BIT CLOCK UNTIL PROM STROBE SETS
5709  024314                            10$:
5710
5711  024314  012760  055401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)     ;LOAD RMMR1
5712
5713  024322  012760  051401  000024            MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5714
5715  024330  016005  000024                    MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
```

```
5716  024334  032705  0C0040          BIT     #WC,R5
5717  024340  001765                  BEQ     10$
5718                          ;SET UP DATA SEQUENCER TO LOCATION 11. WHERE PLFS WILL SET
5719  024342  012704  000260          MOV     #176.,R4                ;MAX NUMBER OF BIT CLOCK
5720  024346                        20$:
5721
5722  024346  012760  055401  000024  MOV     #DMD!MUR!DBEN!MOC!DTO!MCLK,RMMR1(R0)    ;LOAD RMMR1
5723
5724  024354  012760  051401  000024  MOV     #DMD!MUR!DBEN!MOC!DTO,RMMR1(R0) ;LOAD RMMR1
5725
5726  024362  016005  000024          MOV     RMMR1(R0),R5    ,STORE RMMR1 AT R5
5727  024366  032705  002000          BIT     #PLFS,R5                ;EXIT IF PLFS IS SET
5728  024372  001010                  BNE     30$
5729  024374  005304                  DEC     R4                      ;ERROR IF COUNT EXHAUSTED
5730  024376  001363                  BNE     20$
5731  024400  062716  000002          ADD     #2,(SP)                 ;
5732  024404  112776  000035  000000  MOVB    #35,@(SP)               ;CANT SET PLFS
5733  024412  000430                  BR      50$
5734  024414                        30$:
5735                          ;STEP THE MAIN REG CLOCK UNTIL THE TRAILING EDGE OF PROM STORBE
5736                          ; IS DETECTED.
5737
5738  024414  016005  000024          MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
5739  024420  032705  000040          BIT     #WC,R5
5740  024424  001007                  BNE     40$
5741
5742  024426  012760  055401  000024  MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5743
5744  024434  012760  051401  000024  MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5745  024442  000414                  BR      50$
5746  024444                        40$:
5747
5748  024444  016005  000024          MOV     RMMR1(R0),R5    ;STORE RMMR1 AT R5
5749  024450  032705  000040          BIT     #WC,R5
5750  024454  001407                  BEQ     50$
5751
5752  024456  012760  055401  000024  MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
5753
5754  024464  012760  051401  000024  MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
5755  024472  000764                  BR      40$
5756  024474                        50$:
5757  024474  000207                  RTS     PC                      ;EXIT
5758
5759                          .SBTTL  CLOCK SYNC SUBROUTINE
5760
5761                          ;;*****************************************************************
5762
5763                          ;THIS SUBROUTINE WILL SIMULATE THE HEADER AND DATA SYNC
5764                          ;PATTERN BEING READ
5765
5766  024476                  CLKSNC:
5767                          ;VERIFY THAT "PLFS" IS ON
5768
5769  024476  016004  000024          MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
5770  024502  032704  002000          BIT     #PLFS,R4        ;LOOK FOR SYN SET ?
5771  024506  001023                  BNE     10$             ;BRANCH IF SO
```

```
 5772  024510  010437  001142                  MOV     R4,$BDDAT       ;
 5773  024514  042737  175777  001142          BIC     # CPLFS,$BDDAT
 5774  024522  012737  002000  001140          MOV     #PLFS,$GDDAT
 5775  024530  010037  001136                  MOV     R0,$BDADR
 5776  024534  062737  000024  001136          ADD     #RMMR1,$BDADR
 5777  024542  062716  000002                  ADD     #2,(SP)
 5778  024546  112776  000035  000000          MOVB    #35,@(SP)
 5779  024554  000466                          BR      60$
 5780  024556  012705  000021          10$:    MOV     #17.,R5         ;MAKE SURE PROM STROBE IS OFF
 5781  024562  032704  000040          20$:    BIT     #WC,R4
 5782  024566  001434                          BEQ     30$
 5783
 5784  024570  012760  055401  000024          MOV     #MR1AAA!MCLK,RMMR1(R0)  ;LOAD RMMR1
 5785
 5786  024576  012760  051401  000024          MOV     #MR1AAA,RMMR1(R0)       ;LOAD RMMR1
 5787
 5788  024604  016004  000024                  MOV     RMMR1(R0),R4    ;STORE RMMR1 AT R4
 5789  024610  005305                          DEC     R5
 5790  024612  001363                          BNE     20$
 5791                                  ;                       ERROR CAN'T RESET PROM STROBE WITH LFS ACTIVE
 5792  024614  010437  001142                  MOV     R4,$BDDAT       ;SETUP ERROR FOR USER
 5793  024620  042737  177737  001142          BIC     # CWC,$BDDAT
 5794  024626  005037  001140                  CLR     $GDDAT
 5795  024632  010037  001136                  MOV     R0,$BDADR
 5796  024636  062737  000024  001136          ADD     #RMMR1,$BDADR
 5797  024644  062716  000002                  ADD     #2,(SP)
 5798  024650  112776  000062  000000          MOVB    #62,@(SP)
 5799  024656  000425                          BR      60$
 5800  024660                          30$:
 5801                                  ;CLOCK THE SYNC PATTERN (00011001) THROUGH THE SHIFT REGISTER
 5802  024660  012704  014400                  MOV     #014400,R4
 5803  024664  012737  000020  024734          MOV     #16.,70$        ;STROBE BIT COUNT
 5804  024672  012705  055401          40$:    MOV     #MR1AAA!MCLK,R5 ;GENERATE REG VALUE
 5805  024676  000241                          CLC                     ;WITH MAINTENANCE CLOCK ON
 5806  024700  006004                          ROR     R4
 5807  024702  103002                          BCC     50$
 5808  024704  052705  002000                  BIS     #MRD,R5         ;SET READ BIT PER PATTERN BIT
 5809
 5810  024710                          50$:
 5811
 5812  024710  010560  000024                  MOV     R5,RMMR1(R0)    ;LOAD RMMR1
 5813  024714  042705  004000                  BIC     #MCLK,R5                ;CLOCK ONE BIT
 5814
 5815  024720  010560  000024                  MOV     R5,RMMR1(R0)    ;LOAD RMMR1
 5816  024724  005337  024734                  DEC     70$
 5817  024730  001360                          BNE     40$
 5818                                  ;CAN'T VERIFY SYNC CLOCK WAS DETECTED
 5819                                  ;USER CAN DO SO BY STEPING
 5820                                  ;BIT CLOCK AND VERIFY PROM STROBE SETS WITHIN ONE WORD TIME
 5821  024732  000207                  60$:    RTS     PC
 5822  024734  000000                  70$:    .WORD   0               ;TEMPORARY STORAGE
```

```
                                .SBTTL  SAVE AND RESTORE R0-R5 ROUTINES

                                ;;******************************************************************
                                ;*SAVE R0-R5
                                ;*CALL:
                                ;*      SAVREG
                                ;*UPON RETURN FROM $SAVREG THE STACK WILL LOOK LIKE:
                                ;*
                                ;*TOP---(+16)
                                ;* +2---(+18)
                                ;* +4---R5
                                ;* +6---R4
                                ;* +8---R3
                                ;*+10---R2
                                ;*+12---R1
                                ;*+ 4---R0

024736                          $SAVREG:
024736  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
024740  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
024742  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
024744  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
024746  010446                          MOV     R4,-(SP)        ;;PUSH R4 ON STACK
024750  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
024752  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF MAIN FLOW
024756  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF MAIN FLOW
024762  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PS OF CALL
024766  016646  000022                  MOV     22(SP),-(SP)    ;;SAVE PC OF CALL
024772  000002                          RTI

                                ;*RESTORE R0-R5
                                ;*CALL:
                                ;*      RESREG
024774                          $RESREG:
024774  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF CALL
025000  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF CALL
025004  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PC OF MAIN FLOW
025010  012666  000022                  MOV     (SP)+,22(SP)    ;;RESTORE PS OF MAIN FLOW
025014  012605                          MOV     (SP)+,R5        ;;POP STACK INTO R5
025016  012604                          MOV     (SP)+,R4        ;;POP STACK INTO R4
025020  012603                          MOV     (SP)+,R3        ;;POP STACK INTO R3
025022  012602                          MOV     (SP)+,R2        ;;POP STACK INTO R2
025024  012601                          MOV     (SP)+,R1        ;;POP STACK INTO R1
025026  012600                          MOV     (SP)+,R0        ;;POP STACK INTO R0
025030  000002                          RTI
                                .SBTTL  BINARY TO ASCII AND TYPE ROUTINE

                                ;;******************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 16-BIT
                                ;*BINARY-ASCII NUMBER AND TYPE IT.
                                ;*CALL:
                                ;*      MOV     NUMBER,-(SP)    ;;NUMBER TO BE TYPED
                                ;       TYPBN                   ;;TYPE IT

025032  010146                  $TYPBN: MOV     R1,-(SP)        ;;SAVE R1 ON THE STACK
025034  016601  000006                  MOV     6(SP),R1        ;;GET THE INPUT NUMBER
```

```
 5879  025040  000261                          SEC                     ;;SET "C" SO CAN KEEP TRACK OF THE NUMBER OF BITS
 5880  025042  112737  000060  025104  1$:     MOVB    #'0,$BIN        ;;SET CHARACTER TO AN ASCII "0".
 5881  025050  006101                          ROL     R1              ;;GET THIS BIT
 5882  025052  001406                          BEQ     2$              ;;DONE?
 5883  025054  105537  025104                  ADCB    $BIN            ;;NO--SET THE CHARACTER EQUAL TO THIS BIT
 5884  025060  104401  025104                  TYPE    ,$BIN           ;;GO TYPE THIS BIT
 5885  025064  000241                          CLC                     ;;CLEAR "C" SO CAN KEEP TRACK OF BITS
 5886  025066  000765                          BR      1$              ;;GO DO THE NEXT BIT
 5887  025070  012601                  2$:     MOV     (SP)+,R1        ;;POP THE STACK INTO R1
 5888  025072  016666  000002  000004          MOV     2(SP),4(SP)     ;;ADJUST THE STACK
 5889  025100  012616                          MOV     (SP)+,(SP)
 5890  025102  000002                          RTI                     ;;RETURN TO USER
 5891  025104     000     000          $BIN:   .BYTE   0,0             ;;STORAGE FOR ASCII CHAR. AND TERMINATOR
 5892                                  .SBTTL  CONVERT BINARY TO DECIMAL AND TYPE ROUTINE
 5893
 5894                                  ;;**************************************************************
 5895                                  ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 5-DIGIT
 5896                                  ;*SIGNED DECIMAL (ASCII) NUMBER AND TYPE IT. DEPENDING ON WHETHER THE
 5897                                  ;*NUMBER IS POSITIVE OR NEGATIVE A SPACE OR A MINUS SIGN WILL BE TYPED
 5898                                  ;*BEFORE THE FIRST DIGIT OF THE NUMBER. LEADING ZEROS WILL ALWAYS BE
 5899                                  ;*REPLACED WITH SPACES.
 5900                                  ;*CALL:
 5901                                  ;*      MOV     NUM,-(SP)       ;;PUT THE BINARY NUMBER ON THE STACK
 5902                                  ;*      TYPDS                   ;;GO TO THE ROUTINE
 5903
 5904  025106                          $TYPDS:
 5905  025106  010046                          MOV     R0,-(SP)        ;;PUSH R0 ON STACK
 5906  025110  010146                          MOV     R1,-(SP)        ;;PUSH R1 ON STACK
 5907  025112  010246                          MOV     R2,-(SP)        ;;PUSH R2 ON STACK
 5908  025114  010346                          MOV     R3,-(SP)        ;;PUSH R3 ON STACK
 5909  025116  010546                          MOV     R5,-(SP)        ;;PUSH R5 ON STACK
 5910  025120  012746  020200                  MOV     #20200,-(SP)    ;;SET BLANK SWITCH AND SIGN
 5911  025124  016605  000020                  MOV     20(SP),R5       ;;GET THE INPUT NUMBER
 5912  025130  100004                          BPL     1$              ;;BR IF INPUT IS POS.
 5913  025132  005405                          NEG     R5              ;;MAKE THE BINARY NUMBER POS.
 5914  025134  112766  000055  000001          MOVB    #'-,1(SP)       ;;MAKE THE ASCII NUMBER NEG.
 5915  025142  005000                  1$:     CLR     R0              ;;ZERO THE CONSTANTS INDEX
 5916  025144  012703  025322                  MOV     #$DBLK,R3       ;;SETUP THE OUTPUT POINTER
 5917  025150  112723  000040                  MOVB    #' ,(R3)+       ;;SET THE FIRST CHARACTER TO A BLANK
 5918  025154  005002                  2$:     CLR     R2              ;;CLEAR THE BCD NUMBER
 5919  025156  016001  025312                  MOV     $DTBL(R0),R1    ;;GET THE CONSTANT
 5920  025162  160105                  3$:     SUB     R1,R5           ;;FORM THIS BCD DIGIT
 5921  025164  002402                          BLT     4$              ;;BR IF DONE
 5922  025166  005202                          INC     R2              ;;INCREASE THE BCD DIGIT BY 1
 5923  025170  000774                          BR      3$
 5924  025172  060105                  4$:     ADD     R1,R5           ;;ADD BACK THE CONSTANT
 5925  025174  005702                          TST     R2              ;;CHECK IF BCD DIGIT=0
 5926  025176  001002                          BNE     5$              ;;FALL THROUGH IF 0
 5927  025200  105716                          TSTB    (SP)            ;;STILL DOING LEADING 0'S?
 5928  025202  100407                          BMI     7$              ;;BR IF YES
 5929  025204  106316                  5$:     ASLB    (SP)            ;;MSD?
 5930  025206  103003                          BCC     6$              ;;BR IF NO
 5931  025210  116663  000001  177777          MOVB    1(SP),-1(R3)    ;;YES--SET THE SIGN
 5932  025216  052702  000060          6$:     BIS     #'0,R2          ;;MAKE THE BCD DIGIT ASCII
 5933  025222  052702  000040          7$:     BIS     #' ,R2          ;;MAKE IT A SPACE IF NOT ALREADY A DIGIT
 5934  025226  110223                          MOVB    R2,(R3)+        ;;PUT THIS CHARACTER IN THE OUTPUT BUFFER
```

```
                                        TST     (R0)+           ;;JUST INCREMENTING
                                        CMP     R0,#10          ;;CHECK THE TABLE INDEX
                                        BLT     2$              ;;GO DO THE NEXT DIGIT
                                        BGT     8$              ;;GO TO EXIT
                                        MOV     R5,R2           ;;GET THE LSD
                                        BR      6$              ;;GO CHANGE TO ASCII
                                8$:     TSTB    (SP)+           ;;WAS THE LSD THE FIRST NON-ZERO?
                                        BPL     9$              ;;BR IF NO
                                        MOVB    -1(SP),-2(R3)   ;;YES--SET THE SIGN FOR TYPING
                                9$:     CLRB    (R3)            ;;SET THE TERMINATOR
                                        MOV     (SP)+,R5        ;;POP STACK INTO R5
                                        MOV     (SP)+,R3        ;;POP STACK INTO R3
                                        MOV     (SP)+,R2        ;;POP STACK INTO R2
                                        MOV     (SP)+,R1        ;;POP STACK INTO R1
                                        MOV     (SP)+,R0        ;;POP STACK INTO R0
                                        TYPE    ,$DBLK          ;;NOW TYPE THE NUMBER
                                        MOV     2(SP),4(SP)     ;;ADJUST THE STACK
                                        MOV     (SP)+,(SP)
                                        RTI                     ;;RETURN TO USER
                                $DTBL:  10000.
                                        1000.
                                        100.
                                        10.
                                $DBLK:  .BLKW   4
                                .SBTTL  BINARY TO OCTAL (ASCII) AND TYPE

                                ;;**************************************************************
                                ;*THIS ROUTINE IS USED TO CHANGE A 16-BIT BINARY NUMBER TO A 6-DIGIT
                                ;*OCTAL (ASCII) NUMBER AND TYPE IT.
                                ;*$TYPOS---ENTER HERE TO SETUP SUPPRESS ZEROS AND NUMBER OF DIGITS TO TYPE
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOS                   ;;CALL FOR TYPEOUT
                                ;*      .BYTE   N               ;;N=1 TO 6 FOR NUMBER OF DIGITS TO TYPE
                                ;*      .BYTE   M               ;;M=1 OR 0
                                ;*                                      ;;1=TYPE LEADING ZEROS
                                ;*                                      ;;0=SUPPRESS LEADING ZEROS
                                ;*
                                ;*$TYPON----ENTER HERE TO TYPE OUT WITH THE SAME PARAMETERS AS THE LAST
                                ;*$TYPOS OR $TYPOC
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPON                   ;;CALL FOR TYPEOUT
                                ;*
                                ;*$TYPOC---ENTER HERE FOR TYPEOUT OF A 16 BIT NUMBER
                                ;*CALL:
                                ;*      MOV     NUM,-(SP)       ;;NUMBER TO BE TYPED
                                ;*      TYPOC                   ;;CALL FOR TYPEOUT

025332  017646  000000         $TYPOS: MOV     @(SP),-(SP)     ;;PICKUP THE MODE
025336  116637  000001  025555         MOVB    1(SP),$OFILL    ;;LOAD ZERO FILL SWITCH
025344  112637  025557                 MOVB    (SP)+,$OMODE+1  ;;NUMBER OF DIGITS TO TYPE
025350  062716  000002                 ADD     #2,(SP)         ;;ADJUST RETURN ADDRESS
025354  000406                         BR      $TYPON
025356  112737  000001  025555 $TYPOC: MOVB    #1,$OFILL       ;;SET THE ZERO FILL SWITCH
025364  112737  000006  025557         MOVB    #6,$OMODE+1     ;;SET FOR SIX(6) DIGITS
```

```
 5991  025372  112737  000005  025554  $TYPON: MOVB    #5,$OCNT        ;;SET THE ITERATION COUNT
 5992  025400  010346                          MOV     R3,-(SP)        ;;SAVE R3
 5993  025402  010446                          MOV     R4,-(SP)        ;;SAVE R4
 5994  025404  010546                          MOV     R5,-(SP)        ;;SAVE R5
 5995  025406  113704  025557                  MOVB    $OMODE+1,R4     ;;GET THE NUMBER OF DIGITS TO TYPE
 5996  025412  005404                          NEG     R4
 5997  025414  062704  000006                  ADD     #6,R4           ;;SUBTRACT IT FOR MAX. ALLOWED
 5998  025420  110437  025556                  MOVB    R4,$OMODE       ;;SAVE IT FOR USE
 5999  025424  113704  025555                  MOVB    $OFILL,R4       ;;GET THE ZERO FILL SWITCH
 6000  025430  016605  000012                  MOV     12(SP),R5       ;;PICKUP THE INPUT NUMBER
 6001  025434  005003                          CLR     R3              ;;CLEAR THE OUTPUT WORD
 6002  025436  006105                  1$:     ROL     R5              ;;ROTATE MSB INTO "C"
 6003  025440  000404                          BR      3$              ;;GO DO MSB
 6004  025442  006105                  2$:     ROL     R5              ;;FORM THIS DIGIT
 6005  025444  006105                          ROL     R5
 6006  025446  006105                          ROL     R5
 6007  025450  010503                          MOV     R5,R3
 6008  025452  006103                  3$:     ROL     R3              ;;GET LSB OF THIS DIGIT
 6009  025454  105337  025556                  DECB    $OMODE          ;;TYPE THIS DIGIT?
 6010  025460  100016                          BPL     7$              ;;BR IF NO
 6011  025462  042703  177770                  BIC     #177770,R3      ;;GET RID OF JUNK
 6012  025466  001002                          BNE     4$              ;;TEST FOR 0
 6013  025470  005704                          TST     R4              ;;SUPPRESS THIS 0?
 6014  025472  001403                          BEQ     5$              ;;BR IF YES
 6015  025474  005204                  4$:     INC     R4              ;;DON'T SUPPRESS ANYMORE 0'S
 6016  025476  052703  000060                  BIS     #'0,R3          ;;MAKE THIS DIGIT ASCII
 6017  025502  052703  000040          5$:     BIS     #' ,R3          ;;MAKE ASCII IF NOT ALREADY
 6018  025506  110337  025552                  MOVB    R3,8$           ;;SAVE FOR TYPING
 6019  025512  104401  025552                  TYPE    ,8$             ;;GO TYPE THIS DIGIT
 6020  025516  105337  025554          7$:     DECB    $OCNT           ;;COUNT BY 1
 6021  025522  003347                          BGT     2$              ;;BR IF MORE TO DO
 6022  025524  002402                          BLT     6$              ;;BR IF DONE
 6023  025526  005204                          INC     R4              ;;INSURE LAST DIGIT ISN'T A BLANK
 6024  025530  000744                          BR      2$              ;;GO DO THE LAST DIGIT
 6025  025532  012605                  6$:     MOV     (SP)+,R5        ;;RESTORE R5
 6026  025534  012604                          MOV     (SP)+,R4        ;;RESTORE R4
 6027  025536  012603                          MOV     (SP)+,R3        ;;RESTORE R3
 6028  025540  016666  000002  000004          MOV     2(SP),4(SP)     ;;SET THE STACK FOR RETURNING
 6029  025546  012616                          MOV     (SP)+,(SP)
 6030  025550  000002                          RTI                     ;;RETURN
 6031  025552     000          8$:     .BYTE   0               ;;STORAGE FOR ASCII DIGIT
 6032  025553     000                  .BYTE   0               ;;TERMINATOR FOR TYPE ROUTINE
 6033  025554     000          $OCNT:  .BYTE   0               ;;OCTAL DIGIT COUNTER
 6034  025555     000          $OFILL: .BYTE   0               ;;ZERO FILL SWITCH
 6035  025556  000000          $OMODE: .WORD   0               ;;NUMBER OF DIGITS TO TYPE
 6036                          .SBTTL  TYPE ROUTINE
 6037
 6038                          ;;******************************************************************
 6039                          ;*ROUTINE TO TYPE ASCIZ MESSAGE. MESSAGE MUST TERMINATE WITH A 0 BYTE.
 6040                          ;*THE ROUTINE WILL INSERT A NUMBER OF NULL CHARACTERS AFTER A LINE FEED.
 6041                          ;*NOTE1:        $NULL CONTAINS THE CHARACTER TO BE USED AS THE FILLER CHARACTER.
 6042                          ;*NOTE2:        $FILLS CONTAINS THE NUMBER OF FILLER CHARACTERS REQUIRED.
 6043                          ;*NOTE3:        $FILLC CONTAINS THE CHARACTER TO FILL AFTER.
 6044                          ;*
 6045                          ;*CALL:
 6046                          ;*1) USING A TRAP INSTRUCTION
```

```
                                        ;*      TYPE    ,MESADR         ;;MESADR IS FIRST ADDRESS OF AN ASCIZ STRING
                                        ;*OR
                                        ;*      TYPE
                                        ;*      MESADR
                                        ;*

025560  105737  001173        $TYPE:  TSTB    $TPFLG          ;;IS THERE A TERMINAL?
025564  100002                        BPL     1$              ;;BR IF YES
025566  000000                        HALT                    ;;HALT HERE IF NO TERMINAL
025570  000430                        BR      3$              ;;LEAVE
025572  010046                1$:     MOV     R0,-(SP)        ;;SAVE R0
025574  017600  000002                MOV     @2(SP),R0       ;;GET ADDRESS OF ASCIZ STRING
025600  122737  000001  001242        CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
025606  001011                        BNE     62$             ;;NO,GO CHECK FOR APT CONSOLE
025610  132737  000100  001243        BITB    #APTSPOOL,$ENVM ;;SPOOL MESSAGE TO APT
025616  001405                        BEQ     62$             ;;NO,GO CHECK FOR CONSOLE
025620  010037  025630                MOV     R0,61$          ;;SETUP MESSAGE ADDRESS FOR APT
025624  004737  030456                JSR     PC,$TTY3        ;;SPOOL MESSAGE TO APT
025630  000000                61$:    .WORD   0               ;;MESSAGE ADDRESS
025632  132737  000040  001243  62$:   BITB    #APTCSUP,$ENVM  ;;APT CONSOLE SUPPRESSED
025640  001003                        BNE     60$             ;;YES,SKIP TYPE OUT
025642  112046                2$:     MOVB    (R0)+,-(SP)     ;;PUSH CHARACTER TO BE TYPED ONTO STACK
025644  001005                        BNE     4$              ;;BR IF IT ISN'T THE TERMINATOR
025646  005726                        TST     (SP)+           ;;IF TERMINATOR POP IT OFF THE STACK
025650  012600                60$:    MOV     (SP)+,R0        ;;RESTORE R0
025652  062716  000002        3$:     ADD     #2,(SP)         ;;ADJUST RETURN PC
025656  000002                        RTI                     ;;RETURN
025660  122716  000011        4$:     CMPB    #HT,(SP)        ;;BRANCH IF <HT>
025664  001430                        BEQ     8$
025666  122716  000200                CMPB    #CRLF,(SP)      ;;BRANCH IF NOT <CRLF>
025672  001006                        BNE     5$
025674  005726                        TST     (SP)+           ;;POP  <CR><LF> EQUIV
025676  104401                        TYPE                    ;;TYPE A CR AND LF
025700  001217                        $CRLF
025702  105037  026036                CLRB    $CHARCNT        ;;CLEAR CHARACTER COUNT
025706  000755                        BR      2$              ;;GET NEXT CHARACTER
025710  004737  025772        5$:     JSR     PC,$TYPEC       ;;GO TYPE THIS CHARACTER
025714  123726  001172        6$:     CMPB    $FILLC,(SP)+    ;;IS IT TIME FOR FILLER CHARS.?
025720  001350                        BNE     2$              ;;IF NO GO GET NEXT CHAR.
025722  013746  001170                MOV     $NULL,-(SP)     ;;GET # OF FILLER CHARS. NEEDED
                                                            ;;AND THE NULL CHAR.
025726  105366  000001        7$:     DECB    1(SP)           ;;DOES A NULL NEED TO BE TYPED?
025732  002770                        BLT     6$              ;;BR IF NO--GO POP THE NULL OFF OF STACK
025734  004737  025772                JSR     PC,$TYPEC       ;;GO TYPE A NULL
025740  105337  026036                DECB    $CHARCNT        ;;DO NOT COUNT AS A COUNT
025744  000770                        BR      7$              ;;LOOP

                                ;HORIZONTAL TAB PROCESSOR

025746  112716  000040        8$:     MOVB    #' ,(SP)        ;;REPLACE TAB WITH SPACE
025752  004737  025772        9$:     JSR     PC,$TYPEC       ;;TYPE A SPACE
025756  132737  000007  026036        BITB    #7,$CHARCNT     ;;BRANCH IF NOT AT
025764  001372                        BNE     9$              ;;TAB STOP
025766  005726                        TST     (SP)+           ;;POP SPACE OFF STACK
025770  000724                        BR      2$              ;;GET NEXT CHARACTER
025772  105777  153166        $TYPEC: TSTB    @$TPS           ;;WAIT UNTIL PRINTER IS READY
```

```
6103  025776  100375                          BPL     $TYPEC
6104  026000  116677  000002  153160          MOVB    2(SP),@$TPB     ;;LOAD CHAR TO BE TYPED INTO DATA REG.
6105  026006  122766  000015  000002          CMPB    #CR,2(SP)       ;;IS CHARACTER A CARRIAGE RETURN?
6106  026014  001003                          BNE     1$              ;;BRANCH IF NO
6107  026016  105037  026036                  CLRB    $CHARCNT        ;;YES--CLEAR CHARACTER COUNT
6108  026022  000406                          BR      $TYPEX          ;;EXIT
6109  026024  122766  000012  000002  1$:     CMPB    #LF,2(SP)       ;;IS CHARACTER A LINE FEED?
6110  026032  001402                          BEQ     $TYPEX          ;;BRANCH IF YES
6111  026034  105227                          INCB    (PC)+           ;;COUNT THE CHARACTER
6112  026036  000000                  $CHARCNT:.WORD  0               ;;CHARACTER COUNT STORAGE
6113  026040  000207                  $TYPEX: RTS     PC
6114
6115                                  .SBTTL  SCOPE HANDLER ROUTINE
6116
6117                                  ;;**************************************************************************
6118                                  ;*THIS ROUTINE CONTROLS THE LOOPING OF SUBTESTS. IT WILL INCREMENT
6119                                  ;*AND LOAD THE TEST NUMBER($TSTNM) INTO THE DISPLAY REG.(DISPLAY<7:0>)
6120                                  ;*AND LOAD THE ERROR FLAG ($ERFLG) INTO DISPLAY<15:08>
6121                                  ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
6122                                  ;*SW14=1        LOOP ON TEST
6123                                  ;*SW11=1        INHIBIT ITERATIONS
6124                                  ;*SW09=1        LOOP ON ERROR
6125                                  ;*SW08=1        LOOP ON TEST IN SWR<7:0>
6126                                  ;*CALL
6127                                  ;*      SCOPE           ;;SCOPE=IOT
6128
6129  026042                          $SCOPE:
6130  026042  104410                          CKSWR                   ;;TEST  FOR CHANGE IN SOFT-SWR
6131  026044  032777  040000  153102  1$:     BIT     #BIT14,@SWR     ;;LOOP ON PRESENT TEST?
6132  026052  001131                          BNE     $OVER           ;;YES IF SW14=1
6133                                  ;#####START OF CODE FOR THE XOR TESTER#####
6134  026054  000416                  $XTSTR: BR      6$              ;;IF RUNNING ON THE "XOR" TESTER CHANGE
6135                                                                  ;;THIS INSTRUCTION TO A "NOP" (NOP=240)
6136  026056  013746  000004                  MOV     @#ERRVEC,-(SP)  ;;SAVE THE CONTENTS OF THE ERROR VECTOR
6137  026062  012737  026102  000004          MOV     #5$,@#ERRVEC    ;;SET FOR TIMEOUT
6138  026070  005737  177060                  TST     @#177060        ;;TIME OUT ON XOR?
6139  026074  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
6140  026100  000500                          BR      $SVLAD          ;;GO TO THE NEXT TEST
6141  026102  022626                  5$:     CMP     (SP)+,(SP)+     ;;CLEAR THE STACK AFTER A TIME OUT
6142  026104  012637  000004                  MOV     (SP)+,@#ERRVEC  ;;RESTORE THE ERROR VECTOR
6143  026110  000440                          BR      7$              ;;LOOP ON THE PRESENT TEST
6144  026112                          6$:;#####END OF CODE FOR THE XOR TESTER#####
6145  026112  032777  000400  153034          BIT     #BIT08,@SWR     ;;LOOP ON SPEC. TEST?
6146  026120  001421                          BEQ     2$              ;;BR IF NO
6147  026122  005046                          CLR     -(SP)           ;;CLEAR A TEMP. LOCATION
6148  026124  117716  153024                  MOVB    @SWR,(SP)       ;;PICKUP THE DESIRED TEST NUMBER
6149  026130  001414                          BEQ     8$              ;;BRANCH IF BAD TEST NUMBER IN SWR
6150  026132  022716  000024                  CMP     #24,(SP)        ;;CHECK THE NUMBER IN THE SWR
6151  026136  002411                          BLT     8$              ;;BRANCH IF TEST NUMBER IS OUT OF RANGE
6152  026140  011637  001116                  MOV     (SP),$TSTNM     ;;UPDATE THE TEST NUMBER
6153  026144  005316                          DEC     (SP)            ;;BACKUP BY ONE
6154  026146  006316                          ASL     (SP)            ;;SCALE THE TEST NUMBER AS AN INDEX
6155  026150  062716  026354                  ADD     #$SW08TBL,(SP)  ;;FORM THE ADDRESS OF TEST POINTER
6156  026154  013637  001122                  MOV     @(SP)+,$LPADR   ;;SET LOOP ADDRESS TO DESIRED TEST
6157  026160  000466                          BR      $OVER           ;;GO LOOP ON THE TEST
6158  026162  005726                  8$:     TST     (SP)+           ;;CLEAN THE BAD TEST NUMBER OFF OF THE STACK
```

```
6159  026164  105737  001117          2$:     TSTB    $ERFLG          ;;HAS AN ERROR OCCURRED?
6160  026170  001421                          BEQ     3$              ;;BR IF NO
6161  026172  123737  001131  001117          CMPB    $ERMAX,$ERFLG   ;;MAX. ERRORS FOR THIS TEST OCCURRED?
6162  026200  101015                          BHI     3$              ;;BR IF NO
6163  026202  032777  001000  152744          BIT     #BIT09,@SWR     ;;LOOP ON ERROR?
6164  026210  001404                          BEQ     4$              ;;BR IF NO
6165  026212  013737  001124  001122  7$:     MOV     $LPERR,$LPADR   ;;SET LOOP ADDRESS TO LAST SCOPE
6166  026220  000446                          BR      $OVER
6167  026222  105037  001117          4$:     CLRB    $ERFLG          ;;ZERO THE ERROR FLAG
6168  026226  005037  001206                  CLR     $TIMES          ;;CLEAR THE NUMBER OF ITERATIONS TO MAKE
6169  026232  000415                          BR      1$              ;;ESCAPE TO THE NEXT TEST
6170  026234  032777  004000  152712  3$:     BIT     #BIT11,@SWR     ;;INHIBIT ITERATIONS?
6171  026242  001011                          BNE     1$              ;;BR IF YES
6172  026244  005737  001230                  TST     $PASS           ;;IF FIRST PASS OF PROGRAM
6173  026250  001406                          BEQ     1$              ;;      INHIBIT ITERATIONS
6174  026252  005237  001120                  INC     $ICNT           ;;INCREMENT ITERATION COUNT
6175  026256  023737  001206  001120          CMP     $TIMES,$ICNT    ;;CHECK THE NUMBER OF ITERATIONS MADE
6176  026264  002024                          BGE     $OVER           ;;BR IF MORE ITERATION REQUIRED
6177  026266  012737  000001  001120  1$:     MOV     #1,$ICNT        ;;REINITIALIZE THE ITERATION COUNTER
6178  026274  013737  026352  001206          MOV     $MXCNT,$TIMES   ;;SET NUMBER OF ITERATIONS TO DO
6179  026302  105237  001116          $SVLAD: INCB    $TSTNM          ;;COUNT TEST NUMBERS
6180  026306  113737  001116  001226          MOVB    $TSTNM,$TESTN   ;;SET TEST NUMBER IN APT MAILBOX
6181  026314  011637  001122                  MOV     (SP),$LPADR     ;;SAVE SCOPE LOOP ADDRESS
6182  026320  011637  001124                  MOV     (SP),$LPERR     ;;SAVE ERROR LOOP ADDRESS
6183  026324  005037  001210                  CLR     $ESCAPE         ;;CLEAR THE ESCAPE FROM ERROR ADDRESS
6184  026330  112737  000001  001131          MOVB    #1,$ERMAX       ;;ONLY ALLOW ONE(1) ERROR ON NEXT TEST
6185  026336  013777  001116  152612  $OVER:  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER
6186  026344  013716  001122                  MOV     $LPADR,(SP)     ;;FUDGE RETURN ADDRESS
6187  026350  000002                          RTI                     ;;FIXES PS
6188  026352  000012                  $MXCNT: 10.                     ;;MAX. NUMBER OF ITERATIONS
6189  026354                          $SW08TBL:
6190  026354  004432                          .WORD   TST1+2          ;;STARTING ADDRESS OF TEST 1
6191  026356  005006                          .WORD   TST2+2          ;;STARTING ADDRESS OF TEST 2
6192  026360  005226                          .WORD   TST3+2          ;;STARTING ADDRESS OF TEST 3
6193  026362  005442                          .WORD   TST4+2          ;;STARTING ADDRESS OF TEST 4
6194  026364  005630                          .WORD   TST5+2          ;;STARTING ADDRESS OF TEST 5
6195  026366  006766                          .WORD   TST6+2          ;;STARTING ADDRESS OF TEST 6
6196  026370  010146                          .WORD   TST7+2          ;;STARTING ADDRESS OF TEST 7
6197  026372  010326                          .WORD   TST10+2         ;;STARTING ADDRESS OF TEST 10
6198  026374  010450                          .WORD   TST11+2         ;;STARTING ADDRESS OF TEST 11
6199  026376  010750                          .WORD   TST12+2         ;;STARTING ADDRESS OF TEST 12
6200  026400  011336                          .WORD   TST13+2         ;;STARTING ADDRESS OF TEST 13
6201  026402  011762                          .WORD   TST14+2         ;;STARTING ADDRESS OF TEST 14
6202  026404  012346                          .WORD   TST15+2         ;;STARTING ADDRESS OF TEST 15
6203  026406  013060                          .WORD   TST16+2         ;;STARTING ADDRESS OF TEST 16
6204  026410  013520                          .WORD   TST17+2         ;;STARTING ADDRESS OF TEST 17
6205  026412  013772                          .WORD   TST20+2         ;;STARTING ADDRESS OF TEST 20
6206  026414  014514                          .WORD   TST21+2         ;;STARTING ADDRESS OF TEST 21
6207  026416  015474                          .WORD   TST22+2         ;;STARTING ADDRESS OF TEST 22
6208  026420  016744                          .WORD   TST23+2         ;;STARTING ADDRESS OF TEST 23
6209  026422  020170                          .WORD   TST24+2         ;;STARTING ADDRESS OF TEST 24
6210                          .SBTTL  ERROR HANDLER ROUTINE
6211
6212                          ;;**************************************************************
6213                          ;*THIS ROUTINE WILL INCREMENT THE ERROR FLAG AND THE ERROR COUNT,
6214                          ;*SAVE THE ERROR ITEM NUMBER AND THE ADDRESS OF THE ERROR CALL
```

```
                                        ;*AND GO TO ERRTYP ON ERROR
                                        ;*THE SWITCH OPTIONS PROVIDED BY THIS ROUTINE ARE:
                                        ;*SW15=1        HALT ON ERROR
                                        ;*SW13=1        INHIBIT ERROR TYPEOUTS
                                        ;*SW10=1        BELL ON ERROR
                                        ;*SW09=1        LOOP ON ERROR
                                        ;*CALL
                                        ;*      ERROR   N       ;;ERROR=EMT AND N=ERROR ITEM NUMBER

026424                                  $ERROR:
026424  104410                                 CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
026426  105237  001117                  7$:     INCB    $ERFLG          ;;SET THE ERROR FLAG
026432  001775                                  BEQ     7$              ;;DON'T LET THE FLAG GO TO ZERO
026434  013777  001116  152514                  MOV     $TSTNM,@DISPLAY ;;DISPLAY TEST NUMBER AND ERROR FLAG
026442  032777  002000  152504                  BIT     #BIT10,@SWR     ;;BELL ON ERROR?
026450  001402                                  BEQ     1$              ;;NO - SKIP
026452  104401  001212                          TYPE    ,$BELL          ;;RING BELL
026456  005237  001126                  1$:     INC     $ERTTL          ;;COUNT THE NUMBER OF ERRORS
026462  011637  001132                          MOV     (SP),$ERRPC     ;;GET ADDRESS OF ERROR INSTRUCTION
026466  162737  000002  001132                  SUB     #2,$ERRPC
026474  117737  152432  001130                  MOVB    @$ERRPC,$ITEMB  ;;STRIP AND SAVE THE ERROR ITEM CODE
026502  032777  020000  152444                  BIT     #BIT13,@SWR     ;;SKIP TYPEOUT IF SET
026510  001004                                  BNE     20$             ;;SKIP TYPEOUTS
026512  004737  021662                          JSR     PC,ERRTYP       ;;GO TO USER ERROR ROUTINE
026516  104401  001217                          TYPE    ,$CRLF
026522                                  20$:
026522  122737  000001  001242                  CMPB    #APTENV,$ENV    ;;RUNNING IN APT MODE
026530  001007                                  BNE     2$              ;;NO,SKIP APT ERROR REPORT
026532  113737  001130  026544                  MOVB    $ITEMB,21$      ;;SET ITEM NUMBER AS ERROR NUMBER
026540  004737  030466                          JSR     PC,$ATY4        ;;REPORT FATAL ERROR TO APT
026544     000                  21$:    .BYTE   0
026545     000                          .BYTE   0
026546  000777                  22$:    BR      22$             ;;APT ERROR LOOP
026550  005777  152400          2$:     TST     @SWR            ;;HALT ON ERROR
026554  100002                          BPL     3$              ;;SKIP IF CONTINUE
026556  000000                          HALT                    ;;HALT ON ERROR!
026560  104410                          CKSWR                   ;;TEST FOR CHANGE IN SOFT-SWR
026562  032777  001000  152364  3$:     BIT     #BIT09,@SWR     ;;LOOP ON ERROR SWITCH SET?
026570  001402                          BEQ     4$              ;;BR IF NO
026572  013716  001124                  MOV     $LPERR,(SP)     ;;FUDGE RETURN FOR LOOPING
026576  005737  001210          4$:     TST     $ESCAPE         ;;CHECK FOR AN ESCAPE ADDRESS
026602  001402                          BEQ     5$              ;;BR IF NONE
026604  013716  001210                  MOV     $ESCAPE,(SP)    ;;FUDGE RETURN ADDRESS FOR ESCAPE
026610                                  5$:
026610  022737  021642  000042                  CMP     #$ENDAD,@#42    ;;ACT-11 AUTO-ACCEPT?
026616  001001                                  BNE     6$              ;;BRANCH IF NO
026620  000000                                  HALT                    ;;YES
026622                                  6$:
026622  000002                                  RTI                     ;;RETURN
                                        .SBTTL  TTY INPUT ROUTINE

                                        ;;**************************************************************
                                        .ENABL  LSB
026624  000000                  $TKCNT: .WORD   0               ;;NUMBER OF ITEMS IN QUEUE
026626  000000                  $TKQIN: .WORD   0               ;;INPUT POINTER
026630  000000                  $TKQOUT: .WORD  0               ;;OUTPUT POINTER
```

```
6271  026632  000001                  $TKQSRT: .BLKB  1               ;;TTY KEYBOARD QUEUE
6272          026633                  $TKQEND=.
6273          026634                  .EVEN
6274
6275                                  ;*TK INITIALIZE ROUTINE
6276                                  ;*THIS ROUTINE WILL INITIALIZE THE TTY KEYBOARD INPUT QUEUE
6277                                  ;*SETUP THE INTERRUPT VECTOR AND TURN ON THE KEYBOARD INTERRUPT
6278                                  ;
6279                                  ;*CALL:
6280                                  ;*      JSR     PC,$TKINT
6281                                  ;*      RETURN
6282                                  ;
6283  026634  005037  026624         $TKINT: CLR     $TKCNT          ;;CLEAR COUNT OF ITEMS IN QUEUE
6284  026640  012737  026632  026626          MOV     #$TKQSRT,$TKQIN ;;MOVE THE STARTING ADDRESS OF THE
6285  026646  013737  026626  026630          MOV     $TKQIN,$TKQOUT  ;;QUEUE INTO THE INPUT & OUTPUT POINTERS.
6286  026654  012737  026704  000060          MOV     #$TKSRV,@#TKVEC ;;INITIALIZE THE KEYBOARD VECTOR
6287  026662  012737  000200  000062          MOV     #200,@#TKVEC+2  ;;"BR" LEVEL 4
6288  026670  005777  152266                  TST     @$TKB           ;;CLEAR DONE FLAG
6289  026674  012777  000100  152256          MOV     #100,@$TKS      ;;ENABLE TTY KEYBOARD INTERRUPT
6290  026702  000207                          RTS     PC              ;;RETURN TO CALLER
6291
6292                                  ;*TK SERVICE ROUTINE
6293                                  ;*THIS ROUTINE WILL SERVICE THE TTY KEYBOARD INTERRUPT
6294                                  ;*BY READING THE CHARACTER FROM THE INPUT BUFFER AND PUTTING
6295                                  ;*IT IN THE QUEUE.
6296                                  ;*IF THE CHARACTER IS A "CONTROL-C" ( C) $TKINT IS CALLED AND
6297                                  ;*UPON RETURN EXIT IS MADE TO THE "CONTROL-C" RESTART ADDRESS (START)
6298                                  ;
6299  026704  117746  152252         $TKSRV: MOVB    @$TKB,-(SP)     ;;PICKUP THE CHARACTER
6300  026710  042716  177600                  BIC     # C177,(SP)    ;;STRIP THE JUNK
6301  026714  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL C?
6302  026720  001007                          BNE     1$              ;;BRANCH IF NO
6303  026722  104401  030020                  TYPE    ,$CNTLC         ;;TYPE A CONTROL-C ( C)
6304  026726  004737  026634                  JSR     PC,$TKINT       ;;INIT THE KEYBOARD
6305  026732  005726                          TST     (SP)+           ;;CLEAN UP STACK
6306  026734  000137  002632                  JMP     START           ;;CONTROL C RESTART
6307  026740  021627  000007          1$:     CMP     (SP),#7         ;;IS IT A CONTROL G?
6308  026744  001004                          BNE     2$              ;;BRANCH IF NO
6309  026746  022737  000176  001154          CMP     #SWREG,SWR      ;;IS SOFT-SWR SELECTED?
6310  026754  001500                          BEQ     6$              ;;GO TO SWR CHANGE
6311
6312  026756                          2$:
6313  026756  022737  000001  026624          CMP     #1,$TKCNT       ;;IS THE QUEUE FULL?
6314  026764  001004                          BNE     3$              ;;BRANCH IF NO
6315  026766  104401  001212                  TYPE    ,$BELL          ;;RING THE TTY BELL
6316  026772  005726                          TST     (SP)+           ;;CLEAN CHARACTER OFF OF STACK
6317  026774  000451                          BR      5$              ;;EXIT
6318  026776  021627  000023          3$:     CMP     (SP),#23        ;;IS IT A CONTROL-S?
6319  027002  001021                          BNE     32$             ;;BRANCH IF NO
6320  027004  005077  152150                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
6321  027010  005726                          TST     (SP)+           ;;CLEAN CHAR OFF STACK
6322  027012  105777  152142          31$:    TSTB    @$TKS           ;;WAIT FOR A CHAR
6323  027016  100375                          BPL     31$             ;;LOOP UNTIL ITS THERE
6324  027020  117746  152136                  MOVB    @$TKB,-(SP)     ;;GET THE CHARACTER
6325  027024  042716  177600                  BIC     # C177,(SP)    ;;MAKE IT 7-BIT ASCII
6326  027030  022627  000021                  CMP     (SP)+,#21       ;;IS IT A CONTROL-Q?
```

```
 6327  027034  001366                          BNE     31$             ;;BRANCH IF NO
 6328  027036  012777  000100  152114          MOV     #100,@$TKS      ;;REENABLE TTY KEYBOARD INTERRUPTS
 6329  027044  000002                          RTI                     ;;RETURN
 6330  027046  005237  026624          32$:    INC     $TKCNT          ;;COUNT THIS CHARACTER
 6331  027052  021627  000140                  CMP     (SP),#140       ;;IS IT UPPER CASE?
 6332  027056  002405                          BLT     4$              ;;BRANCH IF YES
 6333  027060  021627  000175                  CMP     (SP),#175       ;;IS IT A SPECIAL CHAR?
 6334  027064  003002                          BGT     4$              ;;BRANCH IF YES
 6335  027066  042716  000040                  BIC     #40,(SP)        ;;MAKE IT UPPER CASE
 6336  027072  112677  177530          4$:     MOVB    (SP)+,@$TKQIN   ;;AND PUT IT IN QUEUE
 6337  027076  005237  026626                  INC     $TKQIN          ;;UPDATE THE POINTER
 6338  027102  023727  026626  026633          CMP     $TKQIN,#$TKQEND ;;GO OFF THE END?
 6339  027110  001003                          BNE     5$              ;;BRANCH IF NO
 6340  027112  012737  026632  026626          MOV     #$TKQSRT,$TKQIN ;;RESET THE POINTER
 6341  027120  000002                  5$:     RTI                     ;;RETURN
 6342
 6343                                  ;;**********************************************************************
 6344                                  ;*SOFTWARE SWITCH REGISTER CHANGE ROUTINE.
 6345                                  ;*ROUTINE IS ENTERED  FROM THE TRAP HANDLER, AND WILL
 6346                                  ;*SERVICE THE TEST FOR CHANGE IN SOFTWARE SWITCH REGISTER TRAP
 6347                                  ;*CALL WHEN OPERATING IN TTY INTERRUPT MODE.
 6348  027122  022737  000176  001154  $CKSWR: CMP     #SWREG,SWR      ;,IS THE SOFT-SWR SELECTED
 6349  027130  001124                          BNE     15$             ;;EXIT IF NOT
 6350  027132  105777  152022                  TSTB    @$TKS           ;;IS A CHAR WAITING?
 6351  027136  100121                          BPL     15$             ;;IF NOT, EXIT
 6352  027140  117746  152016                  MOVB    @$TKB,-(SP)     ;;YES
 6353  027144  042716  177600                  BIC     # C177,(SP)     ;;MAKE IT  7-BIT ASCII
 6354  027150  021627  000007                  CMP     (SP),#7         ;;IS IT A CONTROL-G?
 6355  027154  001300                          BNE     2$              ;;IF NOT, PUT IT IN THE TTY QUEUE
 6356                                                                  ;;AND EXIT
 6357
 6358                                  ;;**********************************************************************
 6359                                  ;*CONTROL IS PASSED TO THIS POINT FROM EITHER THE TTY INTERRUPT SERVICE
 6360                                  ;*ROUTINE OR FROM THE SOFTWARE SWITCH REGISTER TRAP CALL, AS A RESULT OF A
 6361                                  ;*CONTROL-G BEING TYPED, AND THE SOFTWARE SWITCH REGISTER BEING SELECTED.
 6362  027156  123727  001150  000001  6$:     CMPB    $AUTOB,#1       ;;ARE WE RUNNING IN AUTO-MODE?
 6363  027164  001674                          BEQ     2$              ;;BRANCH IF YES
 6364  027166  005726                          TST     (SP)+           ;;CLEAR CONTROL-G OFF STACK
 6365  027170  004737  026634                  JSR     PC,$TKINT       ;;FLUSH THE TTY INPUT QUEUE
 6366  027174  005077  151760                  CLR     @$TKS           ;;DISABLE TTY KEYBOARD INTERRUPTS
 6367  027200  112737  000001  001151          MOVB    #1,$INTAG       ;;SET INTERRUPT MODE INDICATOR
 6368
 6369  027206  104401  030032                  TYPE    ,$CNTLG         ;;ECHO THE CONTROL-G ( G)
 6370  027212  104401  030037  $GTSWR: TYPE    ,$MSWR          ;;TYPE CURRENT CONTENTS
 6371  027216  013746  000176                  MOV     SWREG,-(SP)     ;;SAVE SWREG FOR TYPEOUT
 6372  027222  104402                          TYPOC                   ;;GO TYPE--OCTAL ASCII(ALL DIGITS)
 6373  027224  104401  030050                  TYPE    ,$MNEW          ;;PROMPT FOR NEW SWR
 6374  027230  005046                  19$:    CLR     -(SP)           ;;CLEAR COUNTER
 6375  027232  005046                          CLR     -(SP)           ;;THE NEW SWR
 6376  027234  105777  151720          7$:     TSTB    @$TKS           ;;CHAR THERE?
 6377  027240  100375                          BPL     7$              ;;IF NOT TRY AGAIN
 6378
 6379  027242  117746  151714                  MOVB    @$TKB,-(SP)     ;;PICK UP CHAR
 6380  027246  042716  177600                  BIC     # C177,(SP)     ;;MAKE IT 7-BIT ASCII
 6381
 6382  027252  021627  000003                  CMP     (SP),#3         ;;IS IT A CONTROL-C?
```

```
6383  027256  001015                        BNE     9$              ;;BRANCH IF NOT
6384  027260  104401  030020                TYPE    ,$CNTLC         ;;YES, ECHO CONTROL-C ( C)
6385  027264  062706  000006                ADD     #6,SP           ;;CLEAN UP STACK
6386  027270  123727  001151  000001        CMPB    $INTAG,#1       ;;REENABLE TTY KEYBOARD INTERRUPTS?
6387  027276  001003                        BNE     8$              ;;BRANCH IF NO
6388  027300  012777  000100  151652        MOV     #100,@$TKS      ;;ALLOW TTY KEYBOARD INTERRUPTS
6389  027306  000137  002632        8$:     JMP     START           ;;CONTROL-C RESTART
6390
6391
6392  027312  021627  000025        9$:     CMP     (SP),#25        ;;IS IT A CONTROL-U?
6393  027316  001005                        BNE     10$             ;;BRANCH IF NOT
6394  027320  104401  030025                TYPE    ,$CNTLU         ;;YES, ECHO CONTROL-U ( U)
6395  027324  062706  000006        20$:    ADD     #6,SP           ;;IGNORE PREVIOUS INPUT
6396  027330  000737                        BR      19$             ;;LET'S TRY IT AGAIN
6397
6398
6399  027332  021627  000015        10$:    CMP     (SP),#15        ;;IS IT A <CR>?
6400  027336  001022                        BNE     16$             ;;BRANCH IF NO
6401  027340  005766  000004                TST     4(SP)           ;;YES, IS IT THE FIRST CHAR?
6402  027344  001403                        BEQ     11$             ;;BRANCH IF YES
6403  027346  016677  000002  151600        MOV     2(SP),@SWR      ;;SAVE NEW SWR
6404  027354  062706  000006        11$:    ADD     #6,SP           ;;CLEAR UP STACK
6405  027360  104401  001217        14$:    TYPE    ,$CRLF          ;;ECHO <CR> AND <LF>
6406  027364  123727  001151  000001        CMPB    $INTAG,#1       ;;RE-ENABLE TTY KBD INTERRUPTS?
6407  027372  001003                        BNE     15$             ;;BRANCH IF NOT
6408  027374  012777  000100  151556        MOV     #100,@$TKS      ;;RE-ENABLE TTY KBD INTERRUPTS
6409  027402  000002        15$:    RTI                             ;;RETURN
6410  027404  004737  025772        16$:    JSR     PC,$TYPEC       ;;ECHO CHAR
6411  027410  021627  000060                CMP     (SP),#60        ;;CHAR < 0?
6412  027414  002420                        BLT     18$             ;;BRANCH IF YES
6413  027416  021627  000067                CMP     (SP),#67        ;;CHAR > 7?
6414  027422  003015                        BGT     18$             ;;BRANCH IF YES
6415  027424  042726  000060                BIC     #60,(SP)+       ;;STRIP-OFF ASCII
6416  027430  005766  000002                TST     2(SP)           ;;IS THIS THE FIRST CHAR
6417  027434  001403                        BEQ     17$             ;;BRANCH IF YES
6418  027436  006316                        ASL     (SP)            ;;NO, SHIFT PRESENT
6419  027440  006316                        ASL     (SP)            ;;   CHAR OVER TO MAKE
6420  027442  006316                        ASL     (SP)            ;;   ROOM FOR NEW ONE.
6421  027444  005266  000002        17$:    INC     2(SP)           ;;KEEP COUNT OF CHAR
6422  027450  056616  177776                BIS     -2(SP),(SP)     ;;SET IN NEW CHAR
6423  027454  000667                        BR      7$              ;;GET THE NEXT ONE
6424  027456  104401  001216        18$:    TYPE    ,$QUES          ;;TYPE ?<CR><LF>
6425  027462  000720                        BR      20$             ;;SIMULATE CONTROL-U
6426                        .DSABL  LSB
6427
6428
6429                        ;;*****************************************************************
6430                        ;*THIS ROUTINE WILL INPUT A SINGLE CHARACTER FROM THE TTY
6431                        ;*CALL:
6432                        ;*      RDCHR                   ;;GET A CHARACTER FROM THE QUEUE
6433                        ;*      RETURN HERE             ;;CHARACTER IS ON THE STACK
6434                        ;*                              ;;WITH PARITY BIT STRIPPED OFF
6435                        ;
6436
6437  027464  011646        $RDCHR: MOV     (SP),-(SP)      ;;PUSH DOWN THE PC AND
6438  027466  016666  000004  000002        MOV     4(SP),2(SP)     ;;THE PS
```

```
6439  027474  005066  000004          CLR     4(SP)           ;;GET READY FOR A CHARACTER
6440  027500  005046                  CLR     -(SP)           ;;PUT NEW PS ON STACK
6441  027502  012746  027510          MOV     #64$,-(SP)      ;;PUT NEW PC ON  STACK
6442  027506  000002                  RTI                     ;;POP NEW PC AND PS
6443  027510                  64$:
6444  027510  005737  026624  1$:     TST     $TKCNT          ;;WAIT ON A CHARACTER
6445  027514  001775                  BEQ     1$
6446  027516  005337  026624          DEC     $TKCNT          ;;DECREMENT THE COUNTER
6447  027522  117766  177102  000004  MOVB    @$TKQOUT,4(SP)  ;;GET ONE CHARACTER
6448  027530  005237  026630          INC     $TKQOUT         ;;UPDATE THE POINTER
6449  027534  023727  026630  026633  CMP     $TKQOUT,#$TKQEND ;;DID IT GO OFF OF THE END?
6450  027542  001003                  BNE     2$              ;;BRANCH IF NO
6451  027544  012737  026632  026630  MOV     #$TKQSRT,$TKQOUT ;;RESET THE POINTER
6452  027552  000002          2$:     RTI                     ;;RETURN
6453                          ;;**************************************************************
6454                          ;*THIS ROUTINE WILL INPUT A STRING FROM THE TTY
6455                          ;*CALL:
6456                          ;*      RDLIN                   ;;INPUT A STRING FROM THE TTY
6457                          ;*      RETURN HERE             ;;ADDRESS OF FIRST CHARACTER WILL BE ON THE STACK
6458                          ;*                              ;;TERMINATOR WILL BE A BYTE OF ALL 0'S
6459
6460  027554  010346          $RDLIN: MOV     R3,-(SP)        ;;SAVE R3
6461  027556  005046                  CLR     -(SP)           ;;CLEAR THE RUBOUT KEY
6462  027560  012703  030010  1$:     MOV     #$TTYIN,R3      ;;GET ADDRESS
6463  027564  022703  030020  2$:     CMP     #$TTYIN+8.,R3   ;;BUFFER FULL?
6464  027570  101456                  BLOS    4$              ;;BR IF YES
6465  027572  104411                  RDCHR                   ;;GO READ ONE CHARACTER FROM THE TTY
6466  027574  112613                  MOVB    (SP)+,(R3)      ;;GET CHARACTER
6467  027576  122713  000177  10$:    CMPB    #177,(R3)       ;;IS IT A RUBOUT
6468  027602  001022                  BNE     5$              ;;BR IF NO
6469  027604  005716                  TST     (SP)            ;;IS THIS THE FIRST RUBOUT?
6470  027606  001007                  BNE     6$              ;;BR IF NO
6471  027610  112737  000134  030006  MOVB    #'\,9$          ;;TYPE A BACK SLASH
6472  027616  104401  030006          TYPE    ,9$
6473  027622  012716  177777          MOV     #-1,(SP)        ;;SET THE RUBOUT KEY
6474  027626  005303          6$:     DEC     R3              ;;BACKUP BY ONE
6475  027630  020327  030010          CMP     R3,#$TTYIN      ;;STACK EMPTY?
6476  027634  103434                  BLO     4$              ;;BR IF YES
6477  027636  111337  030006          MOVB    (R3),9$         ;;SETUP TO TYPEOUT THE DELETED CHAR.
6478  027642  104401  030006          TYPE    ,9$             ;;GO TYPE
6479  027646  000746                  BR      2$              ;;GO READ ANOTHER CHAR.
6480  027650  005716          5$:     TST     (SP)            ;;RUBOUT KEY SET?
6481  027652  001406                  BEQ     7$              ;;BR IF NO
6482  027654  112737  000134  030006  MOVB    #'\,9$          ;;TYPE A BACK SLASH
6483  027662  104401  030006          TYPE    ,9$
6484  027666  005016                  CLR     (SP)            ;;CLEAR THE RUBOUT KEY
6485  027670  122713  000025  7$:     CMPB    #25,(R3)        ;;IS CHARACTER A CTRL U?
6486  027674  001003                  BNE     8$              ;;BR IF NO
6487  027676  104401  030025          TYPE    ,$CNTLU         ;;TYPE A CONTROL "U"
6488  027702  000726                  BR      1$              ;;GO START OVER
6489  027704  122713  000022  8$:     CMPB    #22,(R3)        ;;IS CHARACTER A "R"?
6490  027710  001011                  BNE     3$              ;;BRANCH IF NO
6491  027712  105013                  CLRB    (R3)            ;;CLEAR THE CHARACTER
6492  027714  104401  001217          TYPE    ,$CRLF          ;;TYPE A "CR" & "LF"
6493  027720  104401  030010          TYPE    ,$TTYIN         ;;TYPE THE INPUT STRING
6494  027724  000717                  BR      2$              ;;GO PICKUP ANOTHER CHACTER
```

```
6495  027726  104401  001216          4$:     TYPE    ,$QUES          ;;TYPE A '?'
6496  027732  000712                          BR      1$              ;;CLEAR THE BUFFER AND LOOP
6497  027734  111337  030006          3$:     MOVB    (R3),9$         ;;ECHO THE CHARACTER
6498  027740  104401  030006                  TYPE    ,9$
6499  027744  122723  000015                  CMPB    #15,(R3)+       ;;CHECK FOR RETURN
6500  027750  001305                          BNE     2$              ;;LOOP IF NOT RETURN
6501  027752  105063  177777                  CLRB    -1(R3)          ;;CLEAR RETURN (THE 15)
6502  027756  104401  001220                  TYPE    ,$LF            ;;TYPE A LINE FEED
6503  027762  005726                          TST     (SP)+           ;;CLEAN RUBOUT KEY FROM THE STACK
6504  027764  012603                          MOV     (SP)+,R3        ;;RESTORE R3
6505  027766  011646                          MOV     (SP),-(SP)      ;;ADJUST THE STACK AND PUT ADDRESS OF THE
6506  027770  016666  000004  000002          MOV     4(SP),2(SP)     ;;      FIRST ASCII CHARACTER ON IT
6507  027776  012766  030010  000004          MOV     #$TTYIN,4(SP)
6508  030004  000002                          RTI                     ;;RETURN
6509  030006     000          9$:     .BYTE   0               ;;STORAGE FOR ASCII CHAR. TO TYPE
6510  030007     000                  .BYTE   0               ;;TERMINATOR
6511  030010  000010          $TTYIN: .BLKB   8.              ;;RESERVE 8 BYTES FOR TTY INPUT
6512  030020  041536  005015     000 $CNTLC: .ASCIZ  / C/<15><12>    ;;CONTROL "C"
6513  030025     136  006525  000012 $CNTLU: .ASCIZ  / U/<15><12>    ;;CONTROL "U"
6514  030032  043536  005015     000 $CNTLG: .ASCIZ  / G/<15><12>    ;;CONTROL "G"
6515  030037     015  051412  051127 $MSWR:  .ASCIZ  <15><12>/SWR = /
6516  030044  036440  000040
6517  030050  020040  042516  020127 $MNEW:  .ASCIZ  /  NEW = /
6518  030056  020075     000
6519          030062                  .EVEN
6520                                  .SBTTL  READ AN OCTAL NUMBER FROM THE TTY
6521
6522                                  ;;**************************************************************
6523                                  ;*THIS ROUTINE WILL READ AN OCTAL (ASCII) NUMBER FROM THE TTY AND
6524                                  ;*CHANGE IT TO BINARY.
6525                                  ;*CALL:
6526                                  ;*      RDOCT                   ;;READ AN OCTAL NUMBER
6527                                  ;*      RETURN HERE             ;;LOW ORDER BITS ARE ON TOP OF THE STACK
6528                                  ;*                              ;;HIGH ORDER BITS ARE IN $HIOCT
6529
6530  030062  011646          $RDOCT: MOV     (SP),-(SP)      ;;PROVIDE SPACE FOR THE
6531  030064  016666  000004  000002  MOV     4(SP),2(SP)     ;;INPUT NUMBER
6532  030072  010046                  MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6533  030074  010146                  MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6534  030076  010246                  MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6535  030100  104412          1$:     RDLIN                   ;;READ AN ASCIZ LINE
6536  030102  012600                  MOV     (SP)+,R0        ;;GET ADDRESS OF 1ST CHARACTER
6537  030104  005001                  CLR     R1              ;;CLEAR DATA WORD
6538  030106  005002                  CLR     R2
6539  030110  112046          2$:     MOVB    (R0)+,-(SP)     ;;PICKUP THIS CHARACTER
6540  030112  001412                  BEQ     3$              ;;IF ZERO GET OUT
6541  030114  006301                  ASL     R1              ;;*2
6542  030116  006102                  ROL     R2
6543  030120  006301                  ASL     R1              ;;*4
6544  030122  006102                  ROL     R2
6545  030124  006301                  ASL     R1              ;;*8
6546  030126  006102                  ROL     R2
6547  030130  042716  177770          BIC     # C7,(SP)       ;;STRIP THE ASCII JUNK
6548  030134  062601                  ADD     (SP)+,R1        ;;ADD IN THIS DIGIT
6549  030136  000764                  BR      2$              ;;LOOP
6550  030140  005726          3$:     TST     (SP)+           ;;CLEAN TERMINATOR FROM STACK
```

```
 6551  030142  010166  000012            MOV     R1,12(SP)       ;;SAVE THE RESULT
 6552  030146  010237  030162            MOV     R2,$HIOCT
 6553  030152  012602                    MOV     (SP)+,R2        ;;POP STACK INTO R2
 6554  030154  012601                    MOV     (SP)+,R1        ;;POP STACK INTO R1
 6555  030156  012600                    MOV     (SP)+,R0        ;;POP STACK INTO R0
 6556  030160  000002                    RTI                     ;;RETURN
 6557  030162  000000            $HIOCT: .WORD   0               ;;HIGH ORDER BITS GO HERE
 6558                            .SBTTL  TRAP DECODER
 6559
 6560                            ;;******************************************************************
 6561                            ;*THIS ROUTINE WILL PICKUP THE LOWER BYTE OF THE "TRAP" INSTRUCTION
 6562                            ;*AND USE IT TO INDEX THROUGH THE TRAP TABLE FOR THE STARTING ADDRESS
 6563                            ;*OF THE DESIRED ROUTINE. THEN USING THE ADDRESS OBTAINED IT WILL
 6564                            ;*GO TO THAT ROUTINE.
 6565
 6566  030164  016646  000002    $TRAP:  MOV     2(SP),-(SP)     ;;ASSUME THE STATUS OF
 6567  030170  042716  000020            BIC     #20,(SP)        ;; THE CALLER--DO NOT ALLOW
 6568  030174  012746  030202            MOV     #1$,-(SP)       ;; T-BIT TRAPS
 6569  030200  000002                    RTI                     ;;SET THE NEW STATUS
 6570  030202  010046            1$:     MOV     R0,-(SP)        ;;SAVE R0
 6571  030204  016600  000002            MOV     2(SP),R0        ;;GET TRAP ADDRESS
 6572  030210  005740                    TST     -(R0)           ;;BACKUP BY 2
 6573  030212  111000                    MOVB    (R0),R0         ;;GET RIGHT BYTE OF TRAP
 6574  030214  006300                    ASL     R0              ;;POSITION FOR INDEXING
 6575  030216  016000  030236            MOV     $TRPAD(R0),R0   ;;INDEX TO TABLE
 6576  030222  000200                    RTS     R0              ;;GO TO ROUTINE
 6577
 6578
 6579                            ;;THIS IS USE TO HANDLE THE "GETPRI" MACRO
 6580
 6581  030224  011646            $TRAP2: MOV     (SP),-(SP)      ;;MOVE THE PC DOWN
 6582  030226  016666  000004  000002    MOV     4(SP),2(SP)     ;;MOVE THE PSW DOWN
 6583  030234  000002                    RTI                     ;;RESTORE THE PSW
 6584
 6585                            .SBTTL  TRAP TABLE
 6586
 6587                            ;*THIS TABLE CONTAINS THE STARTING ADDRESSES OF THE ROUTINES,CALLED
 6588                            ;*BY THE "TRAP" INSTRUCTION.
 6589
 6590                            ;       ROUTINE
 6591                            ;       -------
 6592  030236  030224            $TRPAD: .WORD   $TRAP2
 6593  030240  025560                    $TYPE   ;;CALL=TYPE     TRAP+1(104401)  TTY TYPEOUT ROUTINE
 6594  030242  025356                    $TYPOC  ;;CALL=TYPOC    TRAP+2(104402)  TYPE OCTAL NUMBER (WITH LEADING ZEROS)
 6595  030244  025332                    $TYPOS  ;;CALL=TYPOS    TRAP+3(104403)  TYPE OCTAL NUMBER (NO LEADING ZEROS)
 6596  030246  025372                    $TYPON  ;;CALL=TYPON    TRAP+4(104404)  TYPE OCTAL NUMBER (AS PER LAST CALL)
 6597  030250  025106                    $TYPDS  ;;CALL=TYPDS    TRAP+5(104405)  TYPE DECIMAL NUMBER (WITH SIGN)
 6598  030252  025032                    $TYPBN  ;;CALL=TYPBN    TRAP+6(104406)  TYPE BINARY (ASCII) NUMBER
 6599
 6600  030254  027212                    $GTSWR  ;;CALL=GTSWR    TRAP+7(104407)  GET SOFT-SWR SETTING
 6601
 6602  030256  027122                    $CKSWR  ;;CALL=CKSWR    TRAP+10(104410) TEST FOR CHANGE IN SOFT-SWR
 6603  030260  027464                    $RDCHR  ;;CALL=RDCHR    TRAP+11(104411) TTY TYPEIN CHARACTER ROUTINE
 6604  030262  027554                    $RDLIN  ;;CALL=RDLIN    TRAP+12(104412) TTY TYPEIN STRING ROUTINE
 6605  030264  030062                    $RDOCT  ;;CALL=RDOCT    TRAP+13(104413) READ AN OCTAL NUMBER FROM TTY
 6606  030266  024736                    $SAVREG ;;CALL=SAVREG   TRAP+14(104414) SAVE R0-R5 ROUTINE
```

```
6607  030270  024774                          $RESREG ;;CALL=RESREG   TRAP+15(104415) RESTORE R0-R5 ROUTINE
6608                                  .SBTTL  POWER DOWN AND UP ROUTINES
6609
6610                                  ;;*****************************************************************
6611                                  ;POWER DOWN ROUTINE
6612  030272  012737  030432  000024  $PWRDN: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST UP
6613  030300  012737  000340  000026         MOV     #340,@#PWRVEC+2 ;;PRIO:7
6614  030306  010046                       MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6615  030310  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6616  030312  010246                       MOV     R2,-(SP)        ;;PUSH R2 ON STACK
6617  030314  010346                       MOV     R3,-(SP)        ;;PUSH R3 ON STACK
6618  030316  010446                       MOV     R4,-(SP)        ;;PUSH R4 ON STACK
6619  030320  010546                       MOV     R5,-(SP)        ;;PUSH R5 ON STACK
6620  030322  017746  150626               MOV     @SWR,-(SP)      ;;PUSH @SWR ON STACK
6621  030326  010637  030436               MOV     SP,$SAVR6       ;;SAVE SP
6622  030332  012737  030344  000024       MOV     #$PWRUP,@#PWRVEC ;;SET UP VECTOR
6623  030340  000000                       HALT
6624  030342  000776                       BR      .-2             ;;HANG UP
6625
6626                                  ;;*****************************************************************
6627                                  ;POWER UP ROUTINE
6628  030344  012737  030432  000024  $PWRUP: MOV     #$ILLUP,@#PWRVEC ;;SET FOR FAST DOWN
6629  030352  013706  030436               MOV     $SAVR6,SP       ;;GET SP
6630  030356  005037  030436               CLR     $SAVR6          ;;WAIT LOOP FOR THE TTY
6631  030362  005237  030436         1$:   INC     $SAVR6          ;;WAIT FOR THE INC
6632  030366  001375                       BNE     1$              ;;OF  WORD
6633  030370  012677  150560               MOV     (SP)+,@SWR      ;;POP STACK INTO @SWR
6634  030374  012605                       MOV     (SP)+,R5        ;;POP STACK INTO R5
6635  030376  012604                       MOV     (SP)+,R4        ;;POP STACK INTO R4
6636  030400  012603                       MOV     (SP)+,R3        ;;POP STACK INTO R3
6637  030402  012602                       MOV     (SP)+,R2        ;;POP STACK INTO R2
6638  030404  012601                       MOV     (SP)+,R1        ;;POP STACK INTO R1
6639  030406  012600                       MOV     (SP)+,R0        ;;POP STACK INTO R0
6640  030410  012737  030272  000024       MOV     #$PWRDN,@#PWRVEC ;;SET UP THE POWER DOWN VECTOR
6641  030416  012737  000340  000026       MOV     #340,@#PWRVEC+2 ;;PRIO:7
6642  030424  104401                       TYPE                    ;REPORT THE POWER FAILURE
6643  030426  030440                $PWRMG: .WORD   $POWER          ;;POWER FAIL MESSAGE POINTER
6644  030430  000002                       RTI
6645  030432  000000                $ILLUP: HALT                    ;;THE POWER UP SEQUENCE WAS STARTED
6646  030434  000776                       BR      .-2             ;; BEFORE THE POWER DOWN WAS COMPLETE
6647  030436  000000                $SAVR6: 0                       ;;PUT THE SP HERE
6648  030440  005015  047520  042527  $POWER: .ASCIZ  <15><12>"POWER"
6649  030446  000122
6650                                        .EVEN
6651                                  .SBTTL  APT COMMUNICATIONS ROUTINE
6652
6653                                  ;;*****************************************************************
6654  030450  112737  000001  030714  $ATY1:  MOVB    #1,$FFLG        ;;TO REPORT FATAL ERROR
6655  030456  112737  000001  030712  $ATY3:  MOVB    #1,$MFLG        ;;TO TYPE A MESSAGE
6656  030464  000403                       BR      $ATYC
6657  030466  112737  000001  030714  $ATY4:  MOVB    #1,$FFLG        ;;TO ONLY REPORT FATAL ERROR
6658  030474                        $ATYC:
6659  030474  010046                       MOV     R0,-(SP)        ;;PUSH R0 ON STACK
6660  030476  010146                       MOV     R1,-(SP)        ;;PUSH R1 ON STACK
6661  030500  105737  030712               TSTB    $MFLG           ;;SHOULD TYPE A MESSAGE?
6662  030504  001450                       BEQ     5$              ;;IF NOT:  BR
```

```
 6663  030506  122737  000001  001242          CMPB    #APTENV,$ENV     ;;OPERATING UNDER APT?
 6664  030514  001031                          BNE     3$               ;;IF NOT:  BR
 6665  030516  132737  000100  001243          BITB    #APTSPOOL,$ENVM  ;;SHOULD SPOOL MESSAGES?
 6666  030524  001425                          BEQ     3$               ;;IF NOT:  BR
 6667  030526  017600  000004                  MOV     @4(SP),R0        ;;GET MESSAGE ADDR.
 6668  030532  062766  000002  000004          ADD     #2,4(SP)                 ;;BUMP RETURN ADDR.
 6669  030540  005737  001222          1$:     TST     $MSGTYPE         ;;SEE IF DONE W/ LAST XMISSION?
 6670  030544  001375                          BNE     1$               ;;IF NOT:  WAIT
 6671  030546  010037  001236                  MOV     R0,$MSGAD        ;;PUT ADDR IN MAILBOX
 6672  030552  105720                  2$:     TSTB    (R0)+            ;;FIND END OF MESSAGE
 6673  030554  001376                          BNE     2$
 6674  030556  163700  001236                  SUB     $MSGAD,R0        ;;SUB START OF MESSAGE
 6675  030562  006200                          ASR     R0               ;;GET MESSAGE LNGTH IN WORDS
 6676  030564  010037  001240                  MOV     R0,$MSGLGT       ;;PUT LENGTH IN MAILBOX
 6677  030570  012737  000004  001222          MOV     #4,$MSGTYPE      ;;TELL APT TO TAKE MSG.
 6678  030576  000413                          BR      5$
 6679  030600  017637  000004  030624  3$:     MOV     @4(SP),4$        ;;PUT MSG ADDR IN JSR LINKAGE
 6680  030606  062766  000002  000004          ADD     #2,4(SP)                 ;;BUMP RETURN ADDRESS
 6681  030614  013746  177776                  MOV     177776,-(SP)     ;;PUSH 177776 ON STACK
 6682  030620  004737  025560                  JSR     PC,$TYPE         ;;CALL TYPE MACRO
 6683  030624  000000                  4$:     .WORD   0
 6684  030626                          5$:
 6685  030626  105737  030714          10$:    TSTB    $FFLG            ;;SHOULD REPORT FATAL ERROR?
 6686  030632  001416                          BEQ     12$              ;;IF NOT:  BR
 6687  030634  005737  001242                  TST     $ENV             ;;RUNNING UNDER APT?
 6688  030640  001413                          BEQ     12$              ;;IF NOT:  BR
 6689  030642  005737  001222          11$:    TST     $MSGTYPE         ;;FINISHED LAST MESSAGE?
 6690  030646  001375                          BNE     11$              ;;IF NOT:  WAIT
 6691  030650  017637  000004  001224          MOV     @4(SP),$FATAL    ;;GET ERROR #
 6692  030656  062766  000002  000004          ADD     #2,4(SP)                 ;;BUMP RETURN ADDR.
 6693  030664  005237  001222                  INC     $MSGTYPE         ;;TELL APT TO TAKE ERROR
 6694  030670  105037  030714          12$:    CLRB    $FFLG            ;;CLEAR FATAL FLAG
 6695  030674  105037  030713                  CLRB    $LFLG            ;;CLEAR LOG FLAG
 6696  030700  105037  030712                  CLRB    $MFLG            ;;CLEAR MESSAGE FLAG
 6697  030704  012601                          MOV     (SP)+,R1         ;;POP STACK INTO R1
 6698  030706  012600                          MOV     (SP)+,R0         ;;POP STACK INTO R0
 6699  030710  000207                          RTS     PC               ;;RETURN
 6700  030712     000          $MFLG:  .BYTE   0                ;;MESSG. FLAG
 6701  030713     000          $LFLG:  .BYTE   0                ;;LOG FLAG
 6702  030714     000          $FFLG:  .BYTE   0                ;;FATAL FLAG
 6703          030716                  .EVEN
 6704          000200          APTSIZE=200
 6705          000001          APTENV=001
 6706          000100          APTSPOOL=100
 6707          000040          APTCSUP=040
 6708
```

```
 6709                                   .SBTTL  CONSOLE MESSAGES
 6710
 6711                                   .NLIST  BEX
       030716     051                   CLSPRN: .ASCII  a)a
       030717     075     000           EQUALS: .ASCIZ  a=a
       030721     015  025012     000   PROMPT: .ASCIZ  <CR><LF>a*a
       030725     077     000           QSTMRK: .ASCIZ  a?a
       030727                           HELPQST:
       030727     015  052012  050131   .ASCIZ  <CR><LF>aTYPE HELP TEXT (Y OR N)??a
       030763                           UBUSQST:
       030763     015  041412  040510   .ASCIZ  <CR><LF>aCHANGE RM03 UNIBUS ADDRESS OR VECTOR ADDRESS (Y OR N)<CR> ??a
       031062  005015  051525  020105   CNSL00: .ASCIZ  <CR><LF>aUSE SAME DEVICES (Y OR N) ??a
       031121     015  051012  030115   CNSL01: .ASCIZ  <CR><LF>aRM03 BUS ADDRESS (a
       031146  005015  047105  051124   CNSL02: .ASCII  <CR><LF>aENTRY NOT IN I/O PAGEa
       031175     015  040412  042104           .ASCIZ  <CR><LF>aADDRESS MUST BE >160000a
       031227     015  051012  030115   CNSL03: .ASCIZ  <CR><LF>aRM03 VECTOR ADDRESS (a
       031257     015  042412  052116   CNSL04: .ASCII  <CR><LF>aENTRY OUT OF RANGEa
       031303     015  040412  042104           .ASCIZ  <CR><LF>aADDRESS MUST BE <1000a
       031333     015  051012  030115   CNSL05: .ASCIZ  <CR><LF>aRM03 INTERRUPT PRIORITY (a
       031367     015  042412  052116   CNSL06: .ASCIZ  <CR><LF>aENTRY OUT OF RANGEa
       031414                           CNSL07:
       031414  005015  054524  042520   .ASCII  <CR><LF>aTYPE (A) TO TEST ALL DEVICES, OR TYPE DEVICEa
       031472  047040  046525  042502   .ASCII  a NUMBER(S)a
       031504  005015  042524  046522   .ASCIZ  <CR><LF>aTERMINATE INPUT WITH CARRIAGE RETURNa
       031553     015  047012  052117   NOTEX:  .ASCIZ  <CR><LF>/NOT EXIST DRIVE  /
                                        .LIST BEX
 6712
 6713          031600                   .EVEN
```

```
6714
6715  031600                        FNCDTB:                                 ;FUNCTION CODE TABLE
6716
6717  031600  020000                .WORD   OPI                             ;NOP
6718  031602  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (2)
6719  031604  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEEK
6720  031606  130000                .WORD   ATA!OPI!IVC                     ;RECALIBRATE
6721  031610  020000                .WORD   OPI                             ;DRIVE CLEAR
6722  031612  030000                .WORD   OPI!IVC                         ;RELEASE
6723  031614  130000                .WORD   OPI!ATA!IVC                     ;OFFSET
6724  031616  130000                .WORD   OPI!ATA!IVC                     ;RETURN TO CENTERLINE
6725  031620  020000                .WORD   OPI                             ;READ IN PRESET
6726  031622  020000                .WORD   OPI                             ;PACK ACKNOWLEDGE
6727  031624  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (24)
6728  031626  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (26)
6729  031630  132000                .WORD   ATA!OPI!IVC!IAE                 ;SEARCH
6730  031632  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (32)
6731  031634  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (34)
6732  031636  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (36)
6733  031640  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (40)
6734  031642  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (42)
6735  031644  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (44)
6736  031646  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (46)
6737  031650  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK DATA
6738  031652  073300                .WORD   WCE!OPI!IVC!IAE!AOE!HCE!ECH     ;WRITE CHECK HEADER AND DATA
6739  031654  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (54)
6740  031656  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (56)
6741  031660  037200                .WORD   OPI!IVC!WLE!IAE!AOE!HCE         ;WRITE DATA
6742  031662  037000                .WORD   OPI!IVC!WLE!IAE!AOE             ;WRITE HEADER AND DATA
6743  031664  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (64)
6744  031666  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (66)
6745  031670  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ DATA
6746  031672  033300                .WORD   OPI!IVC!IAE!AOE!HCE!ECH         ;READ HEADER AND DATA
6747  031674  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (74)
6748  031676  130001                .WORD   OPI!ATA!ILF!IVC                 ;ILLEGAL FUNCTION (76)
6749
6750  031700      001               ATNTBL: .BYTE   1.
6751  031701      002                       .BYTE   2.
6752  031702      004                       .BYTE   4.
6753  031703      010                       .BYTE   8.
6754  031704      020                       .BYTE   16.
6755  031705      040                       .BYTE   32.
6756  031706      100                       .BYTE   64.
6757  031707      200                       .BYTE   128.
6758
```

```
 6759                                       .EVEN
 6760
 6761  031710  046420  040006  000000  EMT1:   .WORD   EMS300,EMS1,0
 6762  031716  046436  046461  046506  EMT2:   .WORD   EMS301,EMS302,EMS303,EMS1,EMS304
 6763  031724  040006  046517
 6764  031730  051670  051111  051251          .WORD   EMS511,EMS500,EMS501,EMS502,EMS503,0
 6765  031736  051276  051345  000000
 6766  031744  046436  046526  046461  EMT3:   .WORD   EMS301,EMS306,EMS302
 6767  031752  051670  051415  051251          .WORD   EMS511,EMS505,EMS501,EMS502,0
 6768  031760  051276  000000
 6769  031764  046420  046461  046542  EMT4:   .WORD   EMS300,EMS302,EMS307,EMS2
 6770  031772  040057
 6771  031774  051670  051276  051251          .WORD   EMS511,EMS502,EMS501,EMS503,0
 6772  032002  051345  000000
 6773  032006  046436  046603  046620  EMT5:   .WORD   EMS301,EMS310,EMS311
 6774  032014  051670  051276  051251          .WORD   EMS511,EMS502,EMS501,EMS503,EMS504
 6775  032022  051345  051371
 6776  032026  046664  000000                  .WORD   EMS312,0
 6777  032032  046436  046526  046620  EMT6:   .WORD   EMS301,EMS306,EMS311
 6778  032040  051670  051276  051251          .WORD   EMS511,EMS502,EMS501,EMS503,EMS504,0
 6779  032046  051345  051371  000000
 6780  032054  046436  046722  046461  EMT7:   .WORD   EMS301,EMS313,EMS302
 6781  032062  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,EMS504,EMS503,0
 6782  032070  051371  051345  000000
 6783  032076  047030  047051  046753  EMT10:  .WORD   EMS316,EMS317,EMS314
 6784  032104  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6785  032112  000000
 6786  032114  047030  047051  047002  EMT11:  .WORD   EMS316,EMS317,EMS315
 6787  032122  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6788  032130  000000
 6789  032132  047030  047071  046753  EMT12:  .WORD   EMS316,EMS320,EMS314
 6790  032140  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6791  032146  000000
 6792  032150  047030  047071  047002  EMT13:  .WORD   EMS316,EMS320,EMS315
 6793  032156  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6794  032164  000000
 6795  032166  047030  047111  046753  EMT14:  .WORD   EMS316,EMS321,EMS314
 6796  032174  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6797  032202  000000
 6798  032204  047030  047111  047002  EMT15:  .WORD   EMS316,EMS321,EMS315
 6799  032212  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6800  032220  000000
 6801  032222  047030  047131  046753  EMT16:  .WORD   EMS316,EMS322,EMS314
 6802  032230  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6803  032236  000000
 6804  032240  047030  047131  047002  EMT17:  .WORD   EMS316,EMS322,EMS315
 6805  032246  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6806  032254  000000
 6807  032256  044412  047350  047426  EMT20:  .WORD   EMS71,EMS335,EMS340,EMS72,EMS377,EMS372
 6808  032264  044471  050450  050331
 6809  032272  051670  051345  000000          .WORD   EMS511,EMS503,0
 6810  032300  044412  047373  047426  EMT21:  .WORD   EMS71,EMS336,EMS340,EMS72,EMS400,EMS372
 6811  032306  044471  050461  050331
 6812  032314  051670  051345  000000          .WORD   EMS511,EMS503,0
 6813  032322  044546  047653  044616  EMT22:  .WORD   EMS73,EMS352,EMS74,EMS402,EMS70,EMS406
 6814  032330  050513  044344  050612
```

```
 6815  032336  051670  051345  051371          .WORD   EMS511,EMS503,EMS504,0
 6816  032344  000000
 6817  032346  044546  047653  044671  EMT23:  .WORD   EMS73,EMS352,EMS75,EMS402,EMS77,EMS406
 6818  032354  050513  045006  050612
 6819  032362  051670  051345  051371          .WORD   EMS511,EMS503,EMS504,0
 6820  032370  000000
 6821  032372  044546  047770  044671  EMT24:  .WORD   EMS73,EMS356,EMS75,EMS402,EMS77,EMS377
 6822  032400  050513  045006  050450
 6823  032406  051670  051345  051371          .WORD   EMS511,EMS503,EMS504,0
 6824  032414  000000
 6825  032416  044546  047350  047426  EMT25:  .WORD   EMS73,EMS335,EMS340,EMS76,EMS411
 6826  032424  044745  050654
 6827  032430  051670  051345  000000          .WORD   EMS511,EMS503,0
 6828  032436  046436  046526  046010  EMT26:  .WORD   EMS301,EMS306,EMS252,EMS253,EMS327,EMS254
 6829  032444  046043  047231  046076
 6830  032452  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,EMS502
 6831  032460  051276
 6832  032462  047237  046753  000000          .WORD   EMS330,EMS314,0
 6833  032470  047412  044546  050645  EMT27:  .WORD   EMS337,EMS73,EMS410,EMS76,EMS411
 6834  032476  044745  050654
 6835  032502  051670  051345  000000          .WORD   EMS511,EMS503,0
 6836  032510  047412  045051  047437  EMT30:  .WORD   EMS337,EMS100,EMS341,EMS101
 6837  032516  045111
 6838  032520  051670  051345  051371          .WORD   EMS511,EMS503,EMS504,0
 6839  032526  000000
 6840  032530  046420  046010          EMT31:  .WORD   EMS300,EMS252
 6841  032534  051670  051251  051345          .WORD   EMS511,EMS501,EMS503,0
 6842  032542  000000
 6843  032544  045051  050265          EMT32:  .WORD   EMS100,EMS370
 6844  032550  051670  051371  000000          .WORD   EMS511,EMS504,0
 6845  032556  045051  050672          EMT33:  .WORD   EMS100,EMS412
 6846  032562  051670  051371  000000          .WORD   EMS511,EMS504,0
 6847  032570  045165  050672          EMT34:  .WORD   EMS102,EMS412
 6848  032574  051670  051345  000000          .WORD   EMS511,EMS503,0
 6849  032602  045165  047373          EMT35:  .WORD   EMS102,EMS336
 6850  032606  051670  051345  000000          .WORD   EMS511,EMS503,0
 6851  032614  045051  047350  047426  EMT36:  .WORD   EMS100,EMS335,EMS340,EMS102,EMS334
 6852  032622  045165  047340
 6853  032626  051670  051371  000000          .WORD   EMS511,EMS504,0
 6854  032634  045051  047350  047426  EMT37:  .WORD   EMS100,EMS335,EMS340,EMS102,EMS377,EMS365
 6855  032642  045165  050450  050165
 6856  032650  050706  045273  050742          .WORD   EMS413,EMS104,EMS415
 6857  032656  051670  051371  000000          .WORD   EMS511,EMS504,0
 6858  032664  045051  047373  047426  EMT40:  .WORD   EMS100,EMS336,EMS340,EMS416,EMS104,EMS415
 6859  032672  050762  045273  050742
 6860  032700  051670  051371  000000          .WORD   EMS511,EMS504,0
 6861  032706  045051  047373  047426  EMT41:  .WORD   EMS100,EMS336,EMS340,EMS73,EMS415,EMS402
 6862  032714  044546  050742  050513
 6863  032722  045165  050461                  .WORD   EMS102,EMS400
 6864  032726  051670  051371  000000          .WORD   EMS511,EMS504,0
 6865  032734  045233  050476  047231  EMT42:  .WORD   EMS103,EMS401,EMS327,EMS370
 6866  032742  050265
 6867  032744  051670  051345  000000          .WORD   EMS511,EMS503,0
 6868  032752  050722  045273  050774  EMT43:  .WORD   EMS414,EMS104,EMS417
 6869  032760  051670  051371  000000          .WORD   EMS511,EMS504,0
 6870  032766  045320  050476  047231  EMT44:  .WORD   EMS105,EMS401,EMS327,EMS370
```

```
 6871  032774  050265
 6872  032776  051670  051345  000000          .WORD   EMS511,EMS503,0
 6873  033004  046420  046043          EMT45:  .WORD   EMS300,EMS253
 6874  033010  051670  051251  051345          .WORD   EMS511,EMS501,EMS503,0
 6875  033016  000000
 6876  033020  045361  050476  050774  EMT46:  .WORD   EMS106,EMS401,EMS417
 6877  033026  051670  051345  000000          .WORD   EMS511,EMS503,0
 6878  033034  050706  045426  051013  EMT47:  .WORD   EMS413,EMS107,EMS420,EMS417
 6879  033042  050774
 6880  033044  051670  051371  000000          .WORD   EMS511,EMS504,0
 6881  033052  045436  050706  050602  EMT50:  .WORD   EMS110,EMS413,EMS405,EMS107
 6882  033060  045426
 6883  033062  051670  051345  000000          .WORD   EMS511,EMS503,0
 6884  033070  047500  045475  051026  EMT51:  .WORD   EMS343,EMS111,EMS421
 6885  033076  051670  051371  000000          .WORD   EMS511,EMS504,0
 6886  033104  050722  050762  045426  EMT52:  .WORD   EMS414,EMS416,EMS107,EMS421
 6887  033112  051026
 6888  033114  051670  051371  000000          .WORD   EMS511,EMS504,0
 6889  033122  047256  040173  046201  EMT53:  .WORD   EMS331,EMS4,EMS256
 6890  033130  051670  051251  000000          .WORD   EMS511,EMS501,0
 6891  033136  045513  050706  050602  EMT54:  .WORD   EMS112,EMS413,EMS405,EMS607
 6892  033144  052204
 6893  033146  051670  051345  000000          .WORD   EMS511,EMS503,0
 6894  033154  045551  050476  047231  EMT55:  .WORD   EMS113,EMS401,EMS327,EMS370
 6895  033162  050265
 6896  033164  051670  051345  000000          .WORD   EMS511,EMS503,0
 6897  033172  045610  050476  050774  EMT56:  .WORD   EMS114,EMS401,EMS417
 6898  033200  051670  051345  000000          .WORD   EMS511,EMS503,0
 6899
 6900  033206  046231  047306          EMT57:  .WORD   EMS257,EMS332
 6901  033212  051670  051470  051251          .WORD   EMS511,EMS506,EMS501,0
 6902  033220  000000
 6903
 6904  033222  040224  047324          EMT60:  .WORD   EMS5,EMS333
 6905  033226  051670  051525  051251          .WORD   EMS511,EMS507,EMS501,0
 6906  033234  000000
 6907
 6908  033236  050706  045656  051013  EMT61:  .WORD   EMS413,EMS115,EMS420,EMS417
 6909  033244  050774
 6910  033246  051670  051371  000000          .WORD   EMS511,EMS504,0
 6911  033254  047571  045051  047437  EMT62:  .WORD   EMS346,EMS100,EMS341,EMS101
 6912  033262  045111
 6913  033264  051670  051371  000000          .WORD   EMS511,EMS504,0
 6914  033272  045551  050302          EMT63:  .WORD   EMS113,EMS371
 6915  033276  051670  051345  000000          .WORD   EMS511,EMS503,0
 6916  033304  050706  045673  047667  EMT64:  .WORD   EMS413,EMS116,EMS353
 6917  033312  051670  051371  000000          .WORD   EMS511,EMS504,0
 6918  033320  051043  047667          EMT65:  .WORD   EMS422,EMS353
 6919  033324  051670  051371  000000          .WORD   EMS511,EMS504,0
 6920
 6921  033332  040545  047373  047446  EMT66:  .WORD   EMS12,EMS336,EMS342
 6922  033340  051670  051251  000000          .WORD   EMS511,EMS501,0
 6923
 6924  033346  046420  040274          EMT67:  .WORD   EMS300,EMS6
 6925  033352  051670  051251                  .WORD   EMS511,EMS501
 6926  033356  040340  047324  000000          .WORD   EMS7,EMS333,0
```

```
 6927
 6928  033364  046420  040274  047231  EMT70:  .WORD   EMS300,EMS6,EMS327,EMS7
 6929  033372  040340
 6930  033374  051670  051251  051371          .WORD   EMS511,EMS501,EMS504,0
 6931  033402  000000
 6932
 6933  033404  040274  047350  047426  EMT71:  .WORD   EMS6,EMS335,EMS340,EMS10,EMS333,EMS342
 6934  033412  040411  047324  047446
 6935  033420  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6936  033426  000000
 6937
 6938  033430  040274  047373  047426  EMT72:  .WORD   EMS6,EMS336,EMS340,EMS10,EMS334,EMS342
 6939  033436  040411  047340  047446
 6940  033444  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6941  033452  000000
 6942
 6943  033454  046436  046265  046506  EMT73:  .WORD   EMS301,EMS260,EMS303,EMS11
 6944  033462  040454
 6945  033464  051670  051251  051276          .WORD   EMS511,EMS501,EMS502,0
 6946  033472  000000
 6947
 6948  033474  047500  047514  047446  EMT74:  .WORD   EMS343,EMS344,EMS342,0
 6949  033502  000000
 6950
 6951  033504  046420  040623          EMT75:  .WORD   EMS300,EMS13
 6952  033510  051670  051345  000000          .WORD   EMS511,EMS503,0
 6953
 6954  033516  047571  040623  047543  EMT76:  .WORD   EMS346,EMS13,EMS345
 6955  033524  051670  051345  000000          .WORD   EMS511,EMS503,0
 6956
 6957  033532  047412  040623  047543  EMT77:  .WORD   EMS337,EMS13,EMS345
 6958  033540  051670  051345  000000          .WORD   EMS511,EMS503,0
 6959
 6960  033546  046436  046722  046076  EMT100: .WORD   EMS301,EMS313,EMS254,EMS347,EMS13
 6961  033554  047607  040623
 6962  033560  051670  051345  000000          .WORD   EMS511,EMS503,0
 6963
 6964  033566  047571  040672  047437  EMT101: .WORD   EMS346,EMS14,EMS341,EMS15
 6965  033574  040737
 6966  033576  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 6967  033604  000000
 6968
 6969  033606  047412  044344          EMT102: .WORD   EMS337,EMS70
 6970  033612  051670  051345  000000          .WORD   EMS511,EMS503,0
 6971  033620  046436  046722  046076  EMT103: .WORD   EMS301,EMS313,EMS254,EMS347,EMS15
 6972  033626  047607  040737
 6973  033632  051670  051345                  .WORD   EMS511,EMS503
 6974  033636  040672  047306  000000          .WORD   EMS14,EMS332,0
 6975
 6976  033644  047571  041076  047437  EMT104: .WORD   EMS346,EMS17,EMS341,EMS16
 6977  033652  041015
 6978  033654  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 6979  033662  000000
 6980
 6981  033664  047412  041076  047437  EMT105: .WORD   EMS337,EMS17,EMS341,EMS16
 6982  033672  041015
```

```
 6983  033674  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 6984  033702  000000
 6985
 6986  033704  046436  046722  046076  EMT106: .WORD   EMS301,EMS313,EMS254,EMS347,EMS16
 6987  033712  047607  041015
 6988  033716  051670  051345                  .WORD   EMS511,EMS503
 6989  033722  041076  047306  000000          .WORD   EMS17,EMS332,0
 6990
 6991  033730  047571  041137  047437  EMT107: .WORD   EMS346,EMS20,EMS341,EMS21
 6992  033736  041203
 6993  033740  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 6994  033746  000000
 6995
 6996  033750  041137  047635  041203  EMT110: .WORD   EMS20,EMS351,EMS21,EMS350,EMS22,EMS315
 6997  033756  047630  041262  047002
 6998  033764  051670  051251  000000          .WORD   EMS511,EMS501,0
 6999
 7000  033772  041137  047340  047630  EMT111: .WORD   EMS20,EMS334,EMS350,EMS22,EMS333
 7001  034000  041262  047324
 7002  034004  051670  051251  000000          .WORD   EMS511,EMS501,0
 7003
 7004  034012  047412  041137  047437  EMT112: .WORD   EMS337,EMS20,EMS341,EMS21
 7005  034020  041203
 7006  034022  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7007  034030  000000
 7008
 7009  034032  047412  041137  047437  EMT113: .WORD   EMS337,EMS20,EMS341,EMS21,EMS350,EMS22,EMS334
 7010  034040  041203  047630  041262
 7011  034046  047340
 7012  034050  051670  051251  000000          .WORD   EMS511,EMS501,0
 7013
 7014  034056  041137  047653  041203  EMT114: .WORD   EMS20,EMS352,EMS21,EMS350,EMS22,EMS333
 7015  034064  047630  041262  047324
 7016  034072  051670  051251  000000          .WORD   EMS511,EMS501,0
 7017
 7018  034100  046436  046722  046076  EMT115: .WORD   EMS301,EMS313,EMS254,EMS347,EMS21
 7019  034106  047607  041203
 7020  034112  051670  051345                  .WORD   EMS511,EMS503
 7021  034116  041137  047306  000000          .WORD   EMS20,EMS332,0
 7022
 7023  034124  047571  041327  047437  EMT116: .WORD   EMS346,EMS23,EMS341,EMS24
 7024  034132  041405
 7025  034134  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7026  034142  000000
 7027
 7028  034144  047412  041327  047437  EMT117: .WORD   EMS337,EMS23,EMS341,EMS24
 7029  034152  041405
 7030  034154  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7031  034162  000000
 7032
 7033  034164  046436  046722  046076  EMT120: .WORD   EMS301,EMS313,EMS254,EMS347,EMS24
 7034  034172  047607  041405
 7035  034176  051670  051345                  .WORD   EMS511,EMS503
 7036  034202  041327  047306  000000          .WORD   EMS23,EMS332,0
 7037
 7038  034210  047571  041464  047437  EMT121: .WORD   EMS346,EMS25,EMS341,EMS26
```

```
 7039  034216  041542
 7040  034220  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7041  034226  000000
 7042
 7043  034230  047412  041464  047437  EMT122: .WORD   EMS337,EMS25,EMS341,EMS26
 7044  034236  041542
 7045  034240  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7046  034246  000000
 7047
 7048  034250  046436  046722  046076  EMT123: .WORD   EMS301,EMS313,EMS254,EMS347,EMS26
 7049  034256  047607  041542
 7050  034262  051670  051345                  .WORD   EMS511,EMS503
 7051  034266  041464  047306  000000          .WORD   EMS25,EMS332,0
 7052
 7053  034274  046420  041621  046542  EMT124: .WORD   EMS300,EMS27,EMS307,EMS2
 7054  034302  040057
 7055  034304  051670  051345  000000          .WORD   EMS511,EMS503,0
 7056
 7057  034312  047256  041621  047667  EMT125: .WORD   EMS331,EMS27,EMS353
 7058  034320  051670  051345                  .WORD   EMS511,EMS503
 7059  034324  041665  047002  000000          .WORD   EMS30,EMS315,0
 7060
 7061  034332  047412  041621  047437  EMT126: .WORD   EMS337,EMS27,EMS341,EMS30
 7062  034340  041665
 7063  034342  051670  051345  000000          .WORD   EMS511,EMS503,0
 7064
 7065  034350  046436  046722  046076  EMT127: .WORD   EMS301,EMS313,EMS254,EMS347,EMS30
 7066  034356  047607  041665
 7067  034362  051670  051345                  .WORD   EMS511,EMS503
 7068  034366  041621  047306  000000          .WORD   EMS27,EMS332,0
 7069
 7070  034374  041745  047713  045706  EMT130: .WORD   EMS31,EMS354,EMS250
 7071  034402  051670  051371  051345          .WORD   EMS511,EMS504,EMS503,0
 7072  034410  000000
 7073
 7074  034412  042016  047713  045706  EMT131: .WORD   EMS32,EMS354,EMS250
 7075  034420  051670  051371  051345          .WORD   EMS511,EMS504,EMS503,0
 7076  034426  000000
 7077
 7078  034430  047746  042076  045706  EMT132: .WORD   EMS355,EMS33,EMS250,EMS341,EMS30
 7079  034436  047437  041665
 7080  034442  051670  051371  000000          .WORD   EMS511,EMS504,0
 7081
 7082  034450  047746  042135  045706  EMT133: .WORD   EMS355,EMS34,EMS250,EMS341,EMS30
 7083  034456  047437  041665
 7084  034462  051670  051371  000000          .WORD   EMS511,EMS504,0
 7085
 7086  034470  047256  040173  046137  EMT134: .WORD   EMS331,EMS4,EMS255
 7087  034476  051670  051371  000000          .WORD   EMS511,EMS504,0
 7088
 7089  034504  042173  050010  050032  EMT135: .WORD   EMS35,EMS357,EMS360,EMS15
 7090  034512  040737
 7091  034514  051670  051251  000000          .WORD   EMS511,EMS501,0
 7092
 7093  034522  046316  050106          EMT136: .WORD   EMS261,EMS362
 7094  034526  051670  051345  000000          .WORD   EMS511,EMS503,0
```

```
 7095
 7096  034534  046420  042235  046542  EMT137: .WORD   EMS300,EMS36,EMS307,EMS2
 7097  034542  040057
 7098  034544  051670  051251  000000          .WORD   EMS511,EMS501,0
 7099
 7100  034552  047746  042265  046137  EMT140: .WORD   EMS355,EMS37,EMS255,EMS341,EMS30
 7101  034560  047437  041665
 7102  034564  051670  051371  000000          .WORD   EMS511,EMS504,0
 7103
 7104  034572  047571  042325  047543  EMT141: .WORD   EMS346,EMS40,EMS345
 7105  034600  051670  051371  000000          .WORD   EMS511,EMS504,0
 7106
 7107  034606  047412  042325  047437  EMT142: .WORD   EMS337,EMS40,EMS341,EMS30
 7108  034614  041665
 7109  034616  051670  051371  000000          .WORD   EMS511,EMS504,0
 7110
 7111  034624  050127  046603  042403  EMT143: .WORD   EMS363,EMS310,EMS41
 7112  034632  051670  051251  000000          .WORD   EMS511,EMS501,0
 7113
 7114  034640  047412  042403  047437  EMT144: .WORD   EMS337,EMS41,EMS341,EMS252,EMS327,EMS253
 7115  034646  046010  047231  046043
 7116  034654  051670  051251  000000          .WORD   EMS511,EMS501,0
 7117
 7118  034662  042403  047713  050144  EMT145: .WORD   EMS41,EMS354,EMS364,EMS252,EMS365,EMS253
 7119  034670  046010  050165  046043
 7120  034676  051670  051251  000000          .WORD   EMS511,EMS501,0
 7121
 7122  034704  046436  046526  042235  EMT146: .WORD   EMS301,EMS306,EMS36
 7123  034712  051670  051251  051345          .WORD   EMS511,EMS501,EMS503,0
 7124  034720  000000
 7125
 7126  034722  050172  042451          EMT147: .WORD   EMS366,EMS42
 7127  034726  051670  051345  000000          .WORD   EMS511,EMS503,0
 7128
 7129  034734  050215  047667  050165  EMT150: .WORD   EMS367,EMS353,EMS365,EMS42,EMS354,EMS3
 7130  034742  042451  047713  040126
 7131  034750  051670  051345  000000          .WORD   EMS511,EMS503,0
 7132
 7133  034756  047412  042235          EMT151: .WORD   EMS337,EMS36
 7134  034762  051670  051251  000000          .WORD   EMS511,EMS501,0
 7135
 7136  034770  042533  047713  042235  EMT152: .WORD   EMS43,EMS354,EMS36
 7137  034776  051670  051251  000000          .WORD   EMS511,EMS501,0
 7138
 7139  035004  050215  047667  050165  EMT153: .WORD   EMS367,EMS353,EMS365,EMS36,EMS370
 7140  035012  042235  050265
 7141  035016  051670  051345  000000          .WORD   EMS511,EMS503,0
 7142
 7143  035024  050215  047667  050165  EMT154: .WORD   EMS367,EMS353,EMS365,EMS36,EMS371
 7144  035032  042235  050302
 7145  035036  051670  051345  000000          .WORD   EMS511,EMS503,0
 7146
 7147  035044  046420  042604  046542  EMT155: .WORD   EMS300,EMS44,EMS307,EMS2
 7148  035052  040057
 7149  035054  051670  051345  000000          .WORD   EMS511,EMS503,0
 7150
```

```
 7151  035062  050215  047667  050165  EMT156: .WORD   EMS367,EMS353,EMS365,EMS44,EMS354,EMS3
 7152  035070  042604  047713  040126
 7153  035076  051670  051345  000000          .WORD   EMS511,EMS503,0
 7154
 7155  035104  046420  042645  046542  EMT157: .WORD   EMS300,EMS45,EMS307,EMS2
 7156  035112  040057
 7157  035114  051670  051345  000000          .WORD   EMS511,EMS503,0
 7158
 7159  035122  050215  047667  050165  EMT160: .WORD   EMS367,EMS353,EMS365,EMS45,EMS354,EMS3
 7160  035130  042645  047713  040126
```

```
 7161  035136  051670  051345  051251          .WORD   EMS511,EMS503,EMS501
 7162  035144  042173  047324  000000          .WORD   EMS35,EMS333,0
 7163
 7164  035152  046420  042722  046542  EMT161: .WORD   EMS300,EMS46,EMS307,EMS2
 7165  035160  040057
 7166  035162  051670  051345  000000          .WORD   EMS511,EMS503,0
 7167
 7168  035170  047412  042722  047667  EMT162: .WORD   EMS337,EMS46,EMS353
 7169  035176  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7170  035204  000000
 7171
 7172  035206  042173  047350  047412  EMT163: .WORD   EMS35,EMS335,EMS337,EMS41,EMS334,EMS372
 7173  035214  042403  047340  050331
 7174  035222  051670  051251  000000          .WORD   EMS511,EMS501,0
 7175
 7176  035230  043004  047350  047412  EMT164: .WORD   EMS47,EMS335,EMS337,EMS41,EMS335,EMS372
 7177  035236  042403  047350  050331
 7178  035244  051670  051251  000000          .WORD   EMS511,EMS501,0
 7179
 7180  035252  047412  042173  047231  EMT165: .WORD   EMS337,EMS35,EMS327,EMS47
 7181  035260  043004
 7182  035262  051670  051251                  .WORD   EMS511,EMS501
 7183  035266  042403  047324  050331          .WORD   EMS41,EMS333,EMS372,0
 7184  035274  000000
 7185
 7186  035276  046420  043004  046542  EMT166: .WORD   EMS300,EMS47,EMS307,EMS2
 7187  035304  040057
 7188  035306  051670  051251  051345          .WORD   EMS511,EMS501,EMS503,0
 7189  035314  000000
 7190
 7191  035316  043044  047350  047426  EMT167: .WORD   EMS50,EMS335,EMS340,EMS36,EMS333
 7192  035324  042235  047324
 7193  035330  051670  051251  051345          .WORD   EMS511,EMS501,EMS503,0
 7194  035336  000000
 7195
 7196  035340  047412  042173          EMT170: .WORD   EMS337,EMS35
 7197  035344  051670  051251  000000          .WORD   EMS511,EMS501,0
 7198
 7199  035352  043044  042135  040126  EMT171: .WORD   EMS50,EMS34,EMS3
 7200  035360  051670  051251  000000          .WORD   EMS511,EMS501,0
 7201
 7202  035366  046436  046526  043113  EMT172: .WORD   EMS301,EMS306,EMS51
 7203  035374  051670  051251  051371          .WORD   EMS511,EMS501,EMS504,0
 7204  035402  000000
 7205
 7206  035404  050215  047667  050165  EMT173: .WORD   EMS367,EMS353,EMS365,EMS47,EMS354,EMS3
 7207  035412  043004  047713  040126
 7208  035420  051670  051251  000000          .WORD   EMS511,EMS501,0
 7209
 7210  035426  046420  045706  047231  EMT174: .WORD   EMS300,EMS250,EMS327,EMS255,EMS327,EMS256
 7211  035434  046137  047231  046201
 7212  035442  047437  051767                  .WORD   EMS341,EMS600
 7213  035446  051670  051251  000000          .WORD   EMS511,EMS501,0
 7214
 7215  035454  046420  046201  047437  EMT175: .WORD   EMS300,EMS256,EMS341,EMS600
 7216  035462  051767
```

```
 7217  035464  051670  051251  000000          .WORD   EMS511,EMS501,0
 7218
 7219  035472  046420  045706  047437  EMT176: .WORD   EMS300,EMS250,EMS341,EMS600
 7220  035500  051767
 7221  035502  051670  051371  000000          .WORD   EMS511,EMS504,0
 7222
 7223  035510  046420  046137  047437  EMT177: .WORD   EMS300,EMS255,EMS341,EMS600
 7224  035516  051767
 7225  035520  051670  051371  000000          .WORD   EMS511,EMS504,0
 7226
 7227  035526  047412  043162  047437  EMT200: .WORD   EMS337,EMS52,EMS341,EMS601
 7228  035534  052017
 7229  035536  051670  051251  000000          .WORD   EMS511,EMS501,0
 7230
 7231  035544  047571  043162  047437  EMT201: .WORD   EMS346,EMS52,EMS341,EMS602
 7232  035552  052037
 7233  035554  051670  051251  000000          .WORD   EMS511,EMS501,0
 7234
 7235  035562  047571  043113  047437  EMT202: .WORD   EMS346,EMS51,EMS341,EMS602
 7236  035570  052037
 7237  035572  051670  051251  000000          .WORD   EMS511,EMS501,0
 7238
 7239  035600  047571  043162  047543  EMT203: .WORD   EMS346,EMS52,EMS345,EMS373,EMS255
 7240  035606  050363  046137
 7241  035612  051670  051371  051251          .WORD   EMS511,EMS504,EMS501,0
 7242  035620  000000
 7243
 7244  035622  047571  043162  047437  EMT204: .WORD   EMS346,EMS52,EMS341,EMS27
 7245  035630  041621
 7246  035632  051670  051371  051251          .WORD   EMS511,EMS504,EMS501,0
 7247  035640  000000
 7248
 7249  035642  043223  047713  040126  EMT205: .WORD   EMS53,EMS354,EMS3
 7250  035650  051670  051345  051276          .WORD   EMS511,EMS503,EMS502,EMS510,0
 7251  035656  051615  000000
 7252
 7253  035662  047412  043264  050367  EMT206: .WORD   EMS337,EMS54,EMS374,EMS250,EMS327,EMS255
 7254  035670  045706  047231  046137
 7255  035676  047231  040126          .WORD   EMS327,EMS3
 7256  035702  051670  051371  051251          .WORD   EMS511,EMS504,EMS501,0
 7257  035710  000000
 7258
 7259  035712  043264  047713  040126  EMT207: .WORD   EMS54,EMS354,EMS3
 7260  035720  051670  051251  000000          .WORD   EMS511,EMS501,0
 7261
 7262  035726  043264  047713  050144  EMT210: .WORD   EMS54,EMS354,EMS364,EMS250
 7263  035734  045706
 7264  035736  051670  051371  000000          .WORD   EMS511,EMS504,0
 7265
 7266  035744  043264  047713  050144  EMT211: .WORD   EMS54,EMS354,EMS364,EMS255
 7267  035752  046137
 7268  035754  051670  051371  000000          .WORD   EMS511,EMS504,0
 7269
 7270  035762  047412  043341          EMT212: .WORD   EMS337,EMS55
 7271  035766  051670  051371  000000          .WORD   EMS511,EMS504,0
 7272
```

```
 7273  035774  043417  047340  047543  EMT213: .WORD   EMS56,EMS334,EMS345,EMS373,EMS262,EMS327,EMS251
 7274  036002  050363  046355  047231
 7275  036010  045744
 7276  036012  051670  051251          .WORD   EMS511,EMS501
 7277  036016  043417  047350  000000  .WORD   EMS56,EMS335,0
 7278
 7279  036024  047412  043417  EMT214: .WORD   EMS337,EMS56
 7280  036030  051670  051251  000000  .WORD   EMS511,EMS501,0
 7281
 7282  036036  043512  047324  047630  EMT215: .WORD   EMS57,EMS333,EMS350,EMS60,EMS334
 7283  036044  043576  047340
 7284  036050  051670  051251  000000  .WORD   EMS511,EMS501,0
 7285
 7286  036056  050127  046526  040224  EMT216: .WORD   EM^363,EMS306,EMS5
 7287  036064  051670  051251  000000  .WORD   EMS511,EMS501,0
 7288
 7289  036072  050127  046526  043650  EMT217: .WORD   EMS363,EMS306,EMS61
 7290  036100  051670  051251  000000  .WORD   EMS511,EMS501,0
 7291
 7292  036106  043723  047324  050420  EMT220: .WORD   EMS62,EMS333,EMS375,EMS251
 7293  036114  045744
 7294  036116  051670  051251  000000  .WORD   EMS511,EMS501,0
 7295
 7296  036124  043723  047324  050434  EMT221: .WORD   EMS62,EMS333,EMS376,EMS262
 7297  036132  046355
 7298  036134  051670  051251  000000  .WORD   EMS511,EMS501,0
 7299
 7300  036142  043723  047324  050434  EMT222: .WORD   EMS62,EMS333,EMS376,EMS250
 7301  036150  045706
 7302  036152  051670  051251  000000  .WORD   EMS511,EMS501,0
 7303
 7304  036160  047571  043723  047437  EMT223: .WORD   EMS346,EMS62,EMS341,EMS603
 7305  036166  052100
 7306  036170  051670  051251  000000  .WORD   EMS511,EMS501,0
 7307
 7308  036176  047571  044004  050434  EMT224: .WORD   EMS346,EMS63,EMS376,EMS262
 7309  036204  046355
 7310  036206  051670  051251  051345  .WORD   EMS511,EMS501,EMS503,0
 7311  036214  000000
 7312
 7313  036216  044004  047324  047630  EMT225: .WORD   EMS63,EMS333,EMS350,EMS363,EMS310,EMS262
 7314  036224  050127  046603  046355
 7315  036232  051670  051251  000000  .WORD   EMS511,EMS501,0
 7316
 7317  036240  044004  050010  042235  EMT226: .WORD   EMS63,EMS357,EMS36,EMS372
 7318  036246  050331
 7319  036250  051670  051251  000000  .WORD   EMS511,EMS501,0
 7320
 7321  036256  044004  047770  050032  EMT227: .WORD   EMS63,EMS356,EMS360,EMS15
 7322  036264  040737
 7323  036266  051670  051251  000000  .WORD   EMS511,EMS501,0
 7324
```

```
 7325  036274  044004  047770  050060  EMT230: .WORD   EMS63,EMS356,EMS361,EMS15
 7326  036302  040737
 7327  036304  051670  051251  000000          .WORD   EMS511,EMS501,0
 7328
 7329  036312  044004  047770  042403  EMT231: .WORD   EMS63,EMS356,EMS41
 7330  036320  051670  051251  000000          .WORD   EMS511,EMS501,0
 7331
 7332  036326  042403  050450  047446  EMT232: .WORD   EMS41,EMS377,EMS342,EMS365,EMS63,EMS332
 7333  036334  050165  044004  047306
 7334  036342  051670  051251  000000          .WORD   EMS511,EMS501,0
 7335
 7336  036350  050215  047667  050165  EMT233: .WORD   EMS367,EMS353,EMS365,EMS63,EMS401
 7337  036356  044004  050476
 7338  036362  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7339  036370  000000
 7340
 7341  036372  041015  050450  050331  EMT234: .WORD   EMS16,EMS377,EMS372,EMS365,EMS64,EMS354,EMS3
 7342  036400  050165  044044  047713
 7343  036406  040126
 7344  036410  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7345  036416  000000
 7346
 7347  036420  046420  044114  046542  EMT235: .WORD   EMS300,EMS65,EMS307,EMS2
 7348  036426  040057
 7349  036430  051670  051276  051345          .WORD   EMS511,EMS502,EMS503,0
 7350  036436  000000
 7351
 7352  036440  043417  050450  050434  EMT236: .WORD   EMS56,EMS377,EMS376,EMS252,EMS372,EMS350
 7353  036446  046010  050331  047630
 7354  036454  044114  050476          .WORD   EMS65,EMS401
 7355  036460  051670  051276  051345          .WORD   EMS511,EMS502,EMS503,EMS501,0
 7356  036466  051251  000000
 7357
 7358  036472  046420  044155  046542  EMT237: .WORD   EMS300,EMS66,EMS307,EMS2
 7359  036500  040057
```

```
 7360  036502  051670  051251  051345          .WORD   EMS511,EMS501,EMS503,0
 7361  036510  000000
 7362
 7363  036512  040737  050461  050331  EMT240: .WORD   EMS15,EMS400,EMS372,EMS350,EMS66,EMS401
 7364  036520  047630  044155  050476
 7365  036526  051670  051345  051251          .WORD   EMS511,EMS503,EMS501,0
 7366  036534  000000
 7367
 7368  036536  044155  047373  047426  EMT241: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS604
 7369  036544  040737  050612  050602
 7370  036552  052121
 7371  036554  051670  051345  000000          .WORD   EMS511,EMS503,0
 7372
 7373  036562  050522  052121  050513  EMT242: .WORD   EMS403,EMS604,EMS402,EMS21,EMS377
 7374  036570  041203  050450
 7375  036574  051670  051345                  .WORD   EMS511,EMS503
 7376  036600  042235  047350  000000          .WORD   EMS36,EMS335,0
 7377
 7378  036606  044155  047373  047426  EMT243: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS604
 7379  036614  041542  050563  050602
 7380  036622  052121
 7381  036624  051670  051345  000000          .WORD   EMS511,EMS503,0
 7382
 7383  036632  044004  050476  050602  EMT244: .WORD   EMS63,EMS401,EMS405,EMS604
 7384  036640  052121
 7385  036642  051670  051345  000000          .WORD   EMS511,EMS503,0
 7386
 7387  036650  042235  050265  050602  EMT245: .WORD   EMS36,EMS370,EMS405,EMS604
 7388  036656  052121
 7389  036660  051670  051345  000000          .WORD   EMS511,EMS503,0
 7390
 7391  036666  050522  052121  050513  EMT246: .WORD   EMS403,EMS604,EMS402,EMS24,EMS377
 7392  036674  041405  050450
 7393  036700  051670  051345                  .WORD   EMS511,EMS503
 7394  036704  042235  047350  000000          .WORD   EMS36,EMS335,0
 7395
 7396  036712  044232  047306  050602  EMT247: .WORD   EMS67,EMS332,EMS405,EMS604
 7397  036720  052121
 7398  036722  051670  051345  000000          .WORD   EMS511,EMS503,0
 7399
 7400  036730  044155  047373  047426  EMT250: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS605
 7401  036736  040737  050612  050602
 7402  036744  052146
 7403  036746  051670  051345  000000          .WORD   EMS511,EMS503,0
 7404
 7405  036754  050522  052146  050513  EMT251: .WORD   EMS403,EMS605,EMS402,EMS21,EMS377
 7406  036762  041203  050450
 7407  036766  051670  051345                  .WORD   EMS511,EMS503
 7408  036772  042235  047350  000000          .WORD   EMS36,EMS335,0
 7409
 7410  037000  044155  047373  047426  EMT252: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS605
 7411  037006  041542  050563  050602
 7412  037014  052146
 7413  037016  051670  051345  000000          .WORD   EMS511,EMS503,0
 7414
 7415  037024  044004  050476  050602  EMT253: .WORD   EMS63,EMS401,EMS405,EMS605
```

```
 7416  037032  052146
 7417  037034  051670  051345  000000          .WORD   EMS511,EMS503,0
 7418
 7419  037042  042235  050265  050602  EMT254: .WORD   EMS36,EMS370,EMS405,EMS605
 7420  037050  052146
 7421  037052  051670  051345  000000          .WORD   EMS511,EMS503,0
 7422
 7423  037060  050522  052146  050513  EMT255: .WORD   EMS403,EMS605,EMS402,EMS24,EMS377
 7424  037066  041405  050450
 7425  037072  051670  051345                  .WORD   EMS511,EMS503
 7426  037076  042235  047350  000000          .WORD   EMS36,EMS335,0
 7427
 7428  037104  044232  047306  050602  EMT256: .WORD   EMS67,EMS332,EMS405,EMS605
 7429  037112  052146
 7430  037114  051670  051345  000000          .WORD   EMS511,EMS503,0
 7431
 7432  037122  044155  047373  047426  EMT257: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS606
 7433  037130  040737  050612  050602
 7434  037136  052164
 7435  037140  051670  051345  000000          .WORD   EMS511,EMS503,0
 7436
 7437  037146  050522  052164  050513  EMT260: .WORD   EMS403,EMS606,EMS402,EMS21,EMS377
 7438  037154  041203  050450
 7439  037160  051670  051345                  .WORD   EMS511,EMS503
 7440  037164  042235  047350  000000          .WORD   EMS36,EMS335,0
 7441
 7442  037172  044155  047373  047426  EMT261: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS606
 7443  037200  041542  050563  050602
 7444  037206  052164
 7445  037210  051670  051345  000000          .WORD   EMS511,EMS503,0
 7446
 7447  037216  044004  050476  050602  EMT262: .WORD   EMS63,EMS401,EMS405,EMS606
 7448  037224  052164
 7449  037226  051670  051345  000000          .WORD   EMS511,EMS503,0
 7450
 7451  037234  042235  050265  050602  EMT263: .WORD   EMS36,EMS370,EMS405,EMS606
 7452  037242  052164
 7453  037244  051670  051345  000000          .WORD   EMS511,EMS503,0
 7454
 7455  037252  050522  052164  050513  EMT264: .WORD   EMS403,EMS606,EMS402,EMS24,EMS377
 7456  037260  041405  050450
 7457  037264  051670  051345                  .WORD   EMS511,EMS503
 7458  037270  042235  047350  000000          .WORD   EMS36,EMS335,0
 7459
 7460  037276  044344  050476  050602  EMT265: .WORD   EMS70,EMS401,EMS405,EMS606
 7461  037304  052164
 7462  037306  051670  051345  000000          .WORD   EMS511,EMS503,0
 7463
 7464  037314  044232  047306  050602  EMT266: .WORD   EMS67,EMS332,EMS405,EMS606
 7465  037322  052164
 7466  037324  051670  051345  000000          .WORD   EMS511,EMS503,0
 7467
 7468  037332  044155  047770  050625  EMT267: .WORD   EMS66,EMS356,EMS407
 7469  037340  051670  051345  000000          .WORD   EMS511,EMS503,0
 7470
 7471  037346  044155  047373  047426  EMT270: .WORD   EMS66,EMS336,EMS340,EMS15,EMS406,EMS405,EMS607
```

```
7472  037354  040737  050612  050602
7473  037362  052204
7474  037364  051670  051345  000000          .WORD   EMS511,EMS503,0
7475
7476  037372  050522  052204  050513  EMT271: .WORD   EMS403,EMS607,EMS402,EMS21,EMS377
7477  037400  041203  050450
7478  037404  051670  051345                  .WORD   EMS511,EMS503
7479  037410  041621  047373  000000          .WORD   EMS27,EMS336,0
7480
7481  037416  041621  050265  050602  EMT272: .WORD   EMS27,EMS370,EMS405,EMS607
7482  037424  052204
7483  037426  051670  051345  000000          .WORD   EMS511,EMS503,0
7484
7485  037434  041621  050302  050602  EMT273: .WORD   EMS27,EMS371,EMS405,EMS607
7486  037442  052204
7487  037444  051670  051345  000000          .WORD   EMS511,EMS503,0
7488
7489  037452  042235  050563  050602  EMT274: .WORD   EMS36,EMS404,EMS405,EMS607
7490  037460  052204
7491  037462  051670  051345  000000          .WORD   EMS511,EMS503,0
7492
7493  037470  043223  050476  050602  EMT275: .WORD   EMS53,EMS401,EMS405,EMS607
7494  037476  052204
7495  037500  051670  051345  051276          .WORD   EMS511,EMS503,EMS502,0
7496  037506  000000
7497
7498  037510  044232  047306  050602  EMT276: .WORD   EMS67,EMS332,EMS405,EMS607
7499  037516  052204
7500  037520  051670  051345  000000          .WORD   EMS511,EMS503,0
7501
7502  037526  044155  047373  047426  EMT277: .WORD   EMS66,EMS336,EMS340,EMS26,EMS404,EMS405,EMS607
7503  037534  041542  050563  050602
7504  037542  052204
7505  037544  051670  051345  000000          .WORD   EMS511,EMS503,0
7506
7507  037552  050522  052204  050513  EMT300: .WORD   EMS403,EMS607,EMS402,EMS24,EMS377
7508  037560  041405  050450
7509  037564  051670  051345                  .WORD   EMS511,EMS503
7510  037570  041621  047373  000000          .WORD   EMS27,EMS336,0
7511
7512  037576  044344  050476  050602  EMT301: .WORD   EMS70,EMS401,EMS405,EMS607
7513  037604  052204
7514  037606  051670  051345  000000          .WORD   EMS511,EMS503,0
7515
7516  037614  052222  000000          EHT1:   .WORD   EH1,0
7517  037620  052301  000000          EHT2:   .WORD   EH2,0
7518  037624  052317  000000          EHT5:   .WORD   EH5,0
7519  037630  052356  000000          EHT7:   .WORD   EH7,0
7520  037634  052375  000000          EHT47:  .WORD   EH47,0
7521  037640  052423  000000          EHT52:  .WORD   EH52,0
7522  037644  052522  000000          EHT57:  .WORD   EH57,0
7523  037650  052621  000000          EHT61:  .WORD   EH61,0
7524  037654  052657  000000          EHT65:  .WORD   EH65,0
7525  037660  052734  000000          EHT71:  .WORD   EH71,0
7526  037664  052310  000000          EHT74:  .WORD   EH3,0
7527  037670  053012  000000          EHT115: .WORD   EH115,0
```

```
 7528  037674  053110  000000          EHT130: .WORD   EH130,0
 7529  037700  053227  000000          EHT132: .WORD   EH132,0
 7530  037704  053326  000000          EHT145: .WORD   EH145,0
 7531  037710  053444  000000          EHT150: .WORD   EH150,0
 7532  037714  053543  000000          EHT213: .WORD   EH213,0
 7533  037720  053641  000000          EHT220: .WORD   EH220,0
 7534
 7535  037724  053700                  EDT1:   .WORD   ED1
 7536  037726  053710                  EDT2:   .WORD   ED2
 7537  037730  053714                  EDT5:   .WORD   ED5
 7538  037732  053722                  EDT47:  .WORD   ED47
 7539  037734  053732                  EDT52:  .WORD   ED52
 7540  037736  053744                  EDT57:  .WORD   ED57
 7541  037740  053756                  EDT61:  .WORD   ED61
 7542  037742  053770                  EDT65:  .WORD   ED65
 7543  037744  054000                  EDT71:  .WORD   ED71
 7544  037746  053710                  EDT74:  .WORD   ED2
 7545  037750  054010                  EDT115: .WORD   ED115
 7546  037752  054022                  EDT130: .WORD   ED130
 7547  037754  054010                  EDT132: .WORD   ED115
 7548  037756  054036                  EDT220: .WORD   ED220
 7549
 7550  037760  054042                  EFT1:   .WORD   EF1
 7551  037762  054045                  EFT2:   .WORD   EF2
 7552  037764  054046                  EFT5:   .WORD   EF5
 7553  037766  054050                  EFT57:  .WORD   EF57
 7554  037770  054042                  EFT65:  .WORD   EF1
 7555  037772  054042                  EFT71:  .WORD   EF1
 7556  037774  054045                  EFT74:  .WORD   EF2
 7557  037776  054050                  EFT115: .WORD   EF57
 7558  040000  054054                  EFT130: .WORD   EF130
 7559  040002  054050                  EFT132: .WORD   EF57
 7560  040004  054046                  EFT220: .WORD   EF5
 7561
 7562                                  .NLIST  BEX
       040006  047516  042516  044530  EMS1:   .ASCIZ  @NONEXISTENT DEVICE "NED" (RMCS2,BIT 12) @
       040057     103  047117  051124  EMS2:   .ASCIZ  @CONTROLLER CLEAR "CLR" (RMCS2,BIT 05) @
       040126  052506  041516  044524  EMS3:   .ASCIZ  @FUNCTION CODE (RMCS1, BITS 01 - 05) @
       040173     125  052516  042523  EMS4:   .ASCIZ  @UNUSED BIT POSITIONS OF @
       040224  042504  044526  042503  EMS5:   .ASCIZ  @DEVICE AVAILABLE "DVA" (RMCS1, BIT 11) @
       040274  040520  052122  054511  EMS6:   .ASCIZ  @PARTIY ERROR "PAR" (RMER1, BIT 03) @
       040340  040504  040524  050040  EMS7:   .ASCIZ  @DATA PARITY ERROR "DPE" (RMER2, BIT 03) @
       040411     120  051101  052111  EMS10:  .ASCIZ  @PARITY TEST "PAT" (RMCS2, BIT 04) @
       040454  040515  051523  052502  EMS11:  .ASCII  @MASSBUS CONTROL BUS PARITY ERROR "MCPE" @
       040524  051050  041515  030523          .ASCIZ  @(RMCS1, BIT 13) @
       040545     111  046114  043505  EMS12:  .ASCIZ  @ILLEGAL REGISTER ERROR "ILR" (RMER1, BIT 01) @
       040623     104  040511  047107  EMS13:  .ASCIZ  @DIAGNOSTIC MODE "DMD" (RMMR1, BIT 00) @
       040672  042515  044504  046525  EMS14:  .ASCIZ  @MEDIUM ON LINE "MOL" (RMDS, BIT 12) @
       040737     115  044501  052116  EMS15:  .ASCIZ  @MAINTENANCE UNIT READY "MUR" (RMMR1, BIT 09) @
       041015     115  044501  052116  EMS16:  .ASCIZ  @MAINTENANCE WRITE PROTECT "MWP" (RMMR1, BIT 03) @
       041076  051127  052111  020105  EMS17:  .ASCIZ  @WRITE LOCK "WRL" (RMDS, BIT 11) @
       041137     104  053105  041511  EMS20:  .ASCIZ  @DEVICE CHECK "DVC" (RMER2, BIT 07) @
       041203     115  044501  052116  EMS21:  .ASCIZ  @MAINTENANCE DRIVE FAULT "MDF" (RMMR1, BIT 06) @
       041262  047125  040523  042506  EMS22:  .ASCIZ  @UNSAFE STATUS "UNS" (RMER1, BIT 14) @
       041327     123  042505  020113  EMS23:  .ASCIZ  @SEEK INCOMPLETE STATUS "SKI" (RMER2, BIT 14) @
       041405     115  044501  052116  EMS24:  .ASCIZ  @MAINTENANCE SEEK ERROR "MSER" (RMMR1, BIT 07) @
```

```
041464  047520  044523  044524  EMS25:  .ASCIZ  @POSITIONING IN PROGRESS "PIP" (RMDS, BIT 13) @
041542  040515  047111  042524  EMS26:  .ASCIZ  @MAINTENANCE ON CYLINDER "MOC" (RMMR1, BIT 08) @
041621     105  042116  047440  EMS27:  .ASCIZ  @END OF BLOCK "EBL" (RMMR1, BIT 13) @
041665     104  040511  047107  EMS30:  .ASCIZ  @DIAGNOSTIC END OF BLOCK "DEBL" (RMMR1, BIT 13) @
041745     114  051501  020124  EMS31:  .ASCIZ  @LAST SECTOR STATUS "LS" (RMMR1, BIT 02) @
042016  040514  052123  051440  EMS32:  .ASCIZ  @LAST SECTOR/TRACK STATUS "LST" (RMMR1, BIT 01) @
042076  042523  052103  051117  EMS33:  .ASCIZ  @SECTOR ADDRESS, BITS 00-04 OF @
042135     124  040522  045503  EMS34:  .ASCIZ  @TRACK ADDRESS, BITS 08-10 OF @
042173     126  046117  046525  EMS35:  .ASCIZ  @VOLUME VALID "VV" (RMDS, BIT 06) @
042235     107  020117  044502  EMS36:  .ASCIZ  @GO BIT (RMCS1, BIT 00) @
042265     103  046131  047111  EMS37:  .ASCIZ  @CYLINDER ADDRESS, BIT 00-09 OF @
042325     114  051501  020124  EMS40:  .ASCIZ  @LAST BLOCK TAKEN STATUS "LBT" (RMDS, BIT 10) @
042403     103  046517  047520  EMS41:  .ASCIZ  @COMPOSITE ERROR "ERR" (RMDS, BIT 14) @
042451     103  046517  040515  EMS42:  .ASCIZ  @COMMAND SEQUENCER TEST BIT "TST" (RMMR2, BIT 12) @
```

```
042533    104  044522  042526  EMS43:  .ASCIZ  @DRIVE READY STATUS "DRY" (RMDS, BIT 07) @
042604 047503  052116  047111  EMS44:  .ASCIZ  @CONTINUE "CONT" (RMMR1, BIT 06) @
042645    111  053116  046101  EMS45:  .ASCIZ  @INVALID COMMAND ERROR "IVC" (RMER2, BIT 12) @
042722 047514  051523  047440  EMS46:  .ASCIZ  @LOSS OF SYSTEM CLOCK ERROR "LSC" (RMER2, BIT 11) @
043004 041517  052503  044520  EMS47:  .ASCIZ  @OCCUPIED "OCC" (RMMR1, BIT 15) @
043044 046111  042514  040507  EMS50:  .ASCIZ  @ILLEGAL FUNCTION "ILF" (RMER1, BIT 0) @
043113    117  043106  042523  EMS51:  .ASCIZ  @OFFSET DIRECTION "OFD" (RMOF, BIT 07) @
043162 043117  051506  052105  EMS52:  .ASCIZ  @OFFSET MODE "OM" (RMDS, BIT 00) @
043223    122  047125  040440  EMS53:  .ASCIZ  @RUN AND GO "RG" (RMMR1, BIT 14) @
043264 047111  040526  044514  EMS54:  .ASCIZ  @INVALID ADDRESS ERROR "IAE" (RMER1, BIT 10) @
043341    101  042104  042522  EMS55:  .ASCIZ  @ADDRESS OVERFLOW ERROR "AOE" (RMER1, BIT 09) @
043417    122  043505  051511  EMS56:  .ASCII  @REGISTER MODIFICATION REFUSED ERROR @
043463    042  046522  021122          .ASCIZ  @"RMR" (RMER1, BIT 02) @
043512 051104  053111  020105  EMS57:  .ASCIZ  @DRIVE REQUEST REQUIRED STATUS "DRQ" (RMDT, BIT 11) @
043576 051120  043517  040522  EMS60:  .ASCIZ  @PROGRAMMABLE STATUS "PGM" (RMDS, BIT 09) @
043650 051104  053111  020105  EMS61:  .ASCIZ  @DRIVE PRESENT STATUS "DPR" (RMDS, BIT 08) @
043723    120  051117  020124  EMS62:  .ASCIZ  @PORT REQUEST FLOP "RQA,RQB" (RMMR2, BITS 15,14) @
044004 052101  042524  052116  EMS63:  .ASCIZ  @ATTENTION "ATA" (RMDS, BIT 15) @
044044 051127  052111  020105  EMS64:  .ASCIZ  @WRITE LOCK ERROR "WLE" (RMER1, BIT 11) @
044114 054105  042503  052120  EMS65:  .ASCIZ  @EXCEPTION "REX" (RMMR1, BIT 12) @
044155    111  053116  046101  EMS66:  .ASCIZ  @INVALID COMMAND ERROR "IVC" (RMER2, BIT 12) @
044232 040524  020107  052502  EMS67:  .ASCIZ  @TAG BUS (RMMR2, BITS 00-09) OR TAG CONTROL @
044306 044514  042516  020123          .ASCIZ  @LINES (RMMR2, BITS 10,11,13) @
044344 042523  051101  044103  EMS70:  .ASCIZ  @SEARCH ENABLE "ESRC" (RMMR1, BIT 11) @

044412 050117  051105  052101  EMS71:  .ASCIZ  @OPERATION INCOMPLETE ERROR "OPI"(RMER1 BIT13) @
044471    104  051511  041101  EMS72:  .ASCIZ  @DISARLE SEARCH TIMEOUT "DTO" (RMMR1 BIT 12) @
044546 051104  053111  020105  EMS73:  .ASCIZ  @DRIVE TIMING ERROR "DTE"(RMER1,BIT 12) @
044616 040515  047111  042524  EMS74:  .ASCIZ  @MAINTENANCE INDEX PULSE "MI"(RMMR1 BIT 2) @
044671    115  044501  052116  EMS75:  .ASCIZ  @MAINTENANCE SECTOR PULSE "MS"(RMMR1 BIT 5) @
044745    106  051117  040515  EMS76:  .ASCIZ  @FORMAT BIT "FMT16" (RMOF,BIT12) @
045006 042523  052103  051117  EMS77:  .ASCIZ  @SECTOR COMPARE "MSC"(RMMR1 BIT 1) @
045051    120  047522  020115  EMS100: .ASCIZ  @PROM STROBE "WC" (RMMR1 BIT 5) @
045111    115  044501  052116  EMS101: .ASCIZ  @MAINTENANCE BIT CLOCK "MCLK" (RMMR1 BIT11) @
045165    114  047517  044513  EMS102: .ASCIZ  @LOOKING FOR SYNC "PLFS"(RMMR1,BIT10) @
045233    127  044522  042524  EMS103: .ASCIZ  @WRITE GATE "BB00" (RMMR2 BIT0) @
045273    110  040505  042504  EMS104: .ASCIZ  @HEADER SYNC PATTERN @
045320 042510  042101  051105  EMS105: .ASCIZ  @HEADER AREA "PHA" (RMMR1 BIT 7) @
045361    105  040516  046102  EMS106: .ASCIZ  @ENABLE CRC OUT "ECRC" (RMMR1 BIT 9) @
045426 042510  042101  051105  EMS107: .ASCIZ  @HEADER @
045436 042522  042101  043440  EMS110: .ASCIZ  @READ GATE "BB01"(RMMR2 BIT 1) @
045475    110  040505  042504  EMS111: .ASCIZ  @HEADER ERROR @
045513    104  052101  020101  EMS112: .ASCIZ  @DATA TIMING SEQUENCER OUTPUT @
045551    104  052101  020101  EMS113: .ASCIZ  @DATA AREA "PDA" (RMMR1 BIT 8) @
045610 047105  041101  042514  EMS114: .ASCIZ  @ENABLE ECC OUT "EECC" (RMMR1, BIT 4) @
045656 041505  020103  040520  EMS115: .ASCIZ  @ECC PATTERN @
045673    105  041503  042440  EMS116: .ASCIZ  @ECC ERROR @
045706 044504  045523  040440  EMS250: .ASCIZ  @DISK ADDRESS REGISTER (RMDA) @
045744 047503  052116  047522  EMS251: .ASCIZ  @CONTROL STATUS REGISTER #1 (RMCS1) @
046010 051105  047522  020122  EMS252: .ASCIZ  @ERROR REGISTER #1 (RMER1) @
046043    105  051122  051117  EMS253: .ASCIZ  @ERROR REGISTER #2 (RMER2) @
046076 040515  047111  042524  EMS254: .ASCIZ  @MAINTENANCE REGISTER #1 (RMMR1) @
046137    104  051505  051111  EMS255: .ASCIZ  @DESIRED CYLINDER REGISTER (RMDC) @
046201    117  043106  042523  EMS256: .ASCIZ  @OFFSET REGISTER (RMOF) @
046231    104  044522  042526  EMS257: .ASCIZ  @DRIVE TYPE REGISTER (RMDT) @
046265    110  046117  044504  EMS260: .ASCIZ  @HOLDING REGISTER (RMHR) @
```

```
046316 042523 044522 046101 EMS261: .ASCIZ  @SERIAL NUMBER REGISTER (RMSN) @
046355    101 052124 047105 EMS262: .ASCIZ  @ATTENTION SUMMARY REGISTER (RMAS) @

046420 040503 047116 052117 EMS300: .ASCIZ  @CANNOT CLEAR @
046436 040503 047116 052117 EMS301: .ASCIZ  @CANNOT WRITE/READ @
046461    101 054516 042040 EMS302: .ASCIZ  @ANY DEVICE REGISTER @
046506 044527 044124 052517 EMS303: .ASCIZ  @WITHOUT @
046517    105 051122 051117 EMS304: .ASCIZ  @ERROR @
046526 020101 047117 020105 EMS306: .ASCIZ  @A ONE FROM @
046542 051525 047111 020107 EMS307: .ASCIZ  @USING MASSBUS INITIALIZE, I.E., @
046603    101 055040 051105 EMS310: .ASCIZ  @A ZERO FROM @
046620 053105 051105 020131 EMS311: .ASCIZ  @EVERY DEVICE REGISTER BIT POSITION @
046664 044124 020105 047506 EMS312: .ASCIZ  @THE FOLLOWING BITS ARE STUCK @
046722 020101 044123 043111 EMS313: .ASCIZ  @A SHIFTING ONE BIT FROM @
046753    101 050120 040505 EMS314: .ASCIZ  @APPEARS STUCK AT ZERO @
047002 050101 042520 051101 EMS315: .ASCIZ  @APPEARS STUCK AT ONE @
047030 042522 044507 052123 EMS316: .ASCIZ  @REGISTER SELECT @
047051    061 024040 026061 EMS317: .ASCIZ  @1 (1,2,4,8,16) @
047071    062 024040 026061 EMS320: .ASCIZ  @2 (1,2,4,8,16) @
047111    064 024040 026061 EMS321: .ASCIZ  @4 (1,2,4,8,16) @
047131    070 024040 026061 EMS322: .ASCIZ  @8 (1,2,4,8,16) @
047151    101 046114 047440 EMS323: .ASCIZ  @ALL ONES FROM @
047170 046101 020114 042532 EMS324: .ASCIZ  @ALL ZEROS FROM @
047210 052101 055040 051105 EMS325: .ASCIZ  @AT ZERO @
047221    101 020124 047117 EMS326: .ASCIZ  @AT ONE @
047231    054 047440 020122 EMS327: .ASCIZ  @, OR @
047237    015 041412 020123 EMS330: .ASCIZ  <CR><LF>@CS MBA CLRL @
047256 040503 047116 052117 EMS331: .ASCIZ  @CANNOT READ ZEROS FROM @
047306 051511 044440 041516 EMS332: .ASCIZ  @IS INCORRECT @
047324 051511 047040 052117 EMS333: .ASCIZ  @IS NOT SET @
047340 051511 051440 052105 EMS334: .ASCIZ  @IS SET @
047350 044123 052517 042114 EMS335: .ASCIZ  @SHOULD NOT BE SET @
047373    123 047510 046125 EMS336: .ASCIZ  @SHOULD BE SET @
047412 040503 047116 052117 EMS337: .ASCIZ  @CANNOT SET @
047426 042502 040503 051525 EMS340: .ASCIZ  @BECAUSE @
047437    125 044523 043516 EMS341: .ASCIZ  @USING @
047446 052504 044522 043516 EMS342: .ASCIZ  @DURING REGISTER TRANSFER @
047500 047125 054105 042520 EMS343: .ASCIZ  @UNEXPECTED @
047514 052502 020123 044524 EMS344: .ASCIZ  @BUS TIMEOUT (04 TRAP) @
047543    102 020131 042522 EMS345: .ASCIZ  @BY REGISTER TRANSFER @
047571    103 047101 047516 EMS346: .ASCIZ  @CANNOT RESET @
047607    127 052111 047510 EMS347: .ASCIZ  @WITHOUT SETTING @
047630 052502 020124    000 EMS350: .ASCIZ  @BUT @
047635    127 051501 051040 EMS351: .ASCIZ  @WAS RESET BY @
047653    127 051501 051440 EMS352: .ASCIZ  @WAS SET BY @
047667    111 020116 044504 EMS353: .ASCIZ  @IN DIAGNOSTIC MODE @
047713    111 020123 047111 EMS354: .ASCIZ  @IS INCORRECT ACCORDING TO @
047746 040503 047116 052117 EMS355: .ASCIZ  @CANNOT INCREMENT @
047770 040527 020123 047516 EMS356: .ASCIZ  @WAS NOT SET BY @
050010 040527 020123 047516 EMS357: .ASCIZ  @WAS NOT RESET BY @
050032 020060 047524 030440 EMS360: .ASCIZ  @0 TO 1 TRANSITION OF @
050060 020061 047524 030040 EMS361: .ASCIZ  @1 TO 0 TRANSITION OF @
050106 051511 044440 041516 EMS362: .ASCIZ  @IS INCONSISTENT @
050127    103 047101 047516 EMS363: .ASCIZ  @CANNOT READ @
050144 042524 052123 050040 EMS364: .ASCIZ  @TEST PATTERN IN @
050165    101 042116 000040 EMS365: .ASCIZ  @AND @
```

```
        050172  040503  047116  052117  EMS366: .ASCIZ  @CANNOT INITIALIZE @
        050215     124  042510  041440  EMS367: .ASCIZ  @THE COMMAND SEQUENCER HAS BEEN CLOCKED @
```

```
050265    122  051505  052105  EMS370: .ASCIZ  @RESET EARLY @
050302 044504  020104  047516  EMS371: .ASCIZ  @DID NOT RESET ON TIME @
050331    104  051125  047111  EMS372: .ASCIZ  @DURING COMMAND EXECUTION @
050363    124  020117     000  EMS373: .ASCIZ  @TO @
050367    127  052111  020110  EMS374: .ASCIZ  @WITH ANY COMBINATION OF @
050420 054502  051040  040505  EMS375: .ASCIZ  @BY READING @
050434 054502  053440  044522  EMS376: .ASCIZ  @BY WRITING @
050450 040527  020123  042523  EMS377: .ASCIZ  @WAS SET @
050461    127  051501  047040  EMS400: .ASCIZ  @WAS NOT SET @
050476 044504  020104  047516  EMS401: .ASCIZ  @DID NOT SET @
050513    127  044510  042514  EMS402: .ASCIZ  @WHILE @
050522 047503  046515  047101  EMS403: .ASCIZ  @COMMAND SEQUENCER DID NOT ABORT @
050563    127  051501  047040  EMS404: .ASCIZ  @WAS NOT RESET @
050602 052504  044522  043516  EMS405: .ASCIZ  @DURING @
050612 040527  020123  042522  EMS406: .ASCIZ  @WAS RESET @
050625    123  040505  041522  EMS407: .ASCIZ  @SEARCH TIMEOUT @
050645    101  052106  051105  EMS410: .ASCIZ  @AFTER @
050654 053440  051501  041440  EMS411: .ASCIZ  @ WAS CHANGED @
050672 051440  052105  042440  EMS412: .ASCIZ  @ SET EARLY @
050706 044440  041516  051117  EMS413: .ASCIZ  @ INCORRECT @
050722 041440  047101  047516  EMS414: .ASCIZ  @ CANNOT DETECT @
050742 053440  051501  051440  EMS415: .ASCIZ  @ WAS SIMULATED @
050762 041440  051117  042522  EMS416: .ASCIZ  @ CORRECT @
050774 042040  051125  047111  EMS417: .ASCIZ  @ DURING WRITE @
051013    107  047105  051105  EMS420: .ASCIZ  @GENERATED @
051026 052504  044522  043516  EMS421: .ASCIZ  @DURING READ @
051043    111  041516  051117  EMS422: .ASCIZ  @INCORRECT DATA TRANSFERRED TO MEMORY @

051111    011  042504  044526  EMS500: .ASCII  @       DEVICE IS NONEXISTENT,@<CR><LF>
051142 042011  053105  041511          .ASCII  @       DEVICE IS SWITCHED TO OTHER PORT@<CR><LF>
051205    011  051124  047101          .ASCIZ  @       TRANSCEIVER ENABLE SWITCH IS OFF@<CR><LF>
051251    011  043111  046440  EMS501: .ASCIZ  @       IF MODULE, M7686,@<CR><LF>
051276 046411  051501  041123  EMS502: .ASCIZ  @       MASSBUS TRANSCEIVER,M5922 OR M5923      @<CR><LF>
051345    011  051503  046440  EMS503: .ASCIZ  @       CS MODULE,M7684,@<CR><LF>
051371    011  051504  046440  EMS504: .ASCIZ  @       DS MODULE,M7685,@<CR><LF>
051415    011  042504  044526  EMS505: .ASCIZ  @       DEVICE IS SWITCHED TO A/B PORT POSITION@<CR><LF>
051470 042011  053105  041511  EMS506: .ASCIZ  @       DEVICE IS NOT AN RM03, OR@<CR><LF>
051525    011  042504  044526  EMS507: .ASCIZ  @       DEVICE IS SWITCHED TO PROGRAMMABLE PORT POSITION, OR@<CR><LF>
051615    011  051501  052523  EMS510: .ASCIZ  @       ASSUMING THE RH CONTROLLER HAS NO FAULT@<CR><LF>
051670 005015  050011  047522  EMS511: .ASCII  <CR><LF>@       PROBABLE FAULT(S):@<CR><LF>
051717    011  047050  052117          .ASCIZ  @       (NOT INCLUDING CABLES OR CONNECTORS)@<CR><LF>

051767    122  040505  020104  EMS600: .ASCIZ  @READ IN PRESET COMMAND @
052017    117  043106  042523  EMS601: .ASCIZ  @OFFSET COMMAND @
052037    122  052105  051125  EMS602: .ASCIZ  @RETURN TO CENTER CENTER COMMAND @
052100 042522  042514  051501  EMS603: .ASCIZ  @RELEASE COMMAND @
052121    122  041505  046101  EMS604: .ASCIZ  @RECALIBRATE COMMAND @
052146 042523  045505  041440  EMS605: .ASCIZ  @SEEK COMMAND @
052164 042523  051101  044103  EMS606: .ASCIZ  @SEARCH COMMAND @
052204 040504  040524  041440  EMS607: .ASCIZ  @DATA COMMAND @

052222 054105  041520  042124  EH1:    .ASCII  @EXPCTD  RECEVD  REGSTR@<CR><LF>
052252 052123  052101  051525          .ASCIZ  @STATUS  STATUS  ADRESS@
052301    040  040502  042523  EH2:    .ASCII  @ BASE@<CR><LF>
052310 042101  042522  051523  EH3:    .ASCIZ  @ADRESS@
052317    105  050130  052103  EH5:    .ASCII  @EXPCTD   STUCK@<CR><LF>
```

```
052337     122  051505  046125            .ASCIZ  @RESULT  BIT(S)@
052356  054105  041520  042124  EH7:      .ASCIZ  @EXPCTD  RECEVD@
052375     040  044506  051522  EH47:     .ASCIZ  @ FIRST  SECOND    CRC@
052423     105  050130  052103  EH52:     .ASCII  @EXPCTD  RECEVD  EXPCTD  RECEVD@<CR><LF>
052463     106  051111  052123            .ASCIZ  @FIRST   FIRST   SECOND  SECOND@
052522  047123  050107  052122  EH57:     .ASCII  @SNGPRT  DULPRT  RECEVD  DRVTYP@<CR><LF>
052562  051104  052126  050131            .ASCIZ  @DRVTYP  DRVTYP  DRVTYP  REGADR@
052621     040  054105  042520  EH61:     .ASCIZ  @ EXPECTED ECC    RECEIVED ECC@
052657     105  050130  052103  EH65:     .ASCII  @EXPCTD  RECEVD  TEST@<CR><LF>
052705     123  040524  052524            .ASCIZ  @STATUS  STATUS  REGSTR@
052734  054105  041520  042124  EH71:     .ASCII  @EXPCTD  RECEVD   TEST@<CR><LF>
052763     123  040524  052524            .ASCIZ  @STATUS  STATUS  PATTRN@
053012  054105  041520  042124  EH115:    .ASCII  @EXPCTD  RECEVD  REGSTR   TEST@<CR><LF>
053051     123  040524  052524            .ASCIZ  @STATUS  STATUS  ADRESS  PATTRN@
053110  054105  041520  042124  EH130:    .ASCII  @EXPCTD  RECEVD  REGSTR   TEST   OFFSET@<CR><LF>
053160  052123  052101  051525            .ASCIZ  @STATUS  STATUS  ADRESS  PATTRN  REGSTR@
053227     105  050130  052103  EH132:    .ASCII  @EXPCTD  ACTUAL  REGSTR  OFFSET@<CR><LF>
053267     103  052517  052116            .ASCIZ  @COUNT   COUNT   ADRESS  REGSTR@
053326  054105  041520  042124  EH145:    .ASCII  @EXPCTD  ACTUAL  REGSTR  RMER1   RMER2@<CR><LF>
053375     103  050115  051105            .ASCIZ  @CMPERR  CMPERR  ADRESS  PATTRN  PATTRN@

053444  054105  041520  042124  EH150:    .ASCII  @EXPCTD  ACTUAL  REGSTR FUNCTION@<CR><LF>
053505     122  051505  046125            .ASCIZ  @RESULT  RESULT  ADRESS   CODE@
053543     105  050130  052103  EH213:    .ASCII  @EXPCTD  ACTUAL  STATUS   TEST@<CR><LF>
053602  042522  052523  052114            .ASCIZ  @RESULT  RESULT  ADRESS  REGSTR@
053641     101  052103  040525  EH220:    .ASCII  @ACTUAL  REGSTR@<CR><LF>
053661     122  051505  046125            .ASCIZ  @RESULT  ADRESS@
                                .LIST BEX
7563                            .EVEN
7564  053700  001140  001142  001136  ED1:      .WORD   $GDDAT,$BDDAT,$BDADR,0
7565  053706  000000
7566  053710  001136  000000          ED2:      .WORD   $BDADR,0
7567  053714  001140  001142  000000  ED5:      .WORD   $GDDAT,$BDDAT,0
7568  053722  001174  001176  001200  ED47:     .WORD   $TMP0,$TMP1,$TMP2,0
7569  053730  000000
7570  053732  001174  054062  001176  ED52:     .WORD   $TMP0,BUFFER,$TMP1,BUFFER+2,0
7571  053740  054064  000000
7572  053744  001174  001176  001142  ED57:     .WORD   $TMP0,$TMP1,$BDDAT,$BDADR,0
7573  053752  001136  000000
7574  053756  001200  001202  001174  ED61:     .WORD   $TMP2,$TMP3,$TMP0,$TMP1,0
7575  053764  001176  000000
7576  053770  001140  001142  001174  ED65:     .WORD   $GDDAT,$BDDAT,$TMP0,0
7577  053776  000000
7578  054000  001140  001142  001440  ED71:     .WORD   $GDDAT,$BDDAT,RMHR0,0
7579  054006  000000
7580  054010  001140  001142  001136  ED115:    .WORD   $GDDAT,$BDDAT,$BDADR,$TMP0,0
7581  054016  001174  000000
7582  054022  001140  001142  001136  ED130:    .WORD   $GDDAT,$BDDAT,$BDADR,$TMP0,$TMP1,0
7583  054030  001174  001176  000000
7584  054036  001142  001136          ED220:    .WORD   $BDDAT,$BDADR
7585
7586  054042     000     000     000  EF1:      .BYTE   0,0,0
7587  054045     000                  EF2:      .BYTE   0
7588  054046     000     000          EF5:      .BYTE   0,0
7589  054050     000     000     000  EF57:     .BYTE   0,0,0,0
7590  054053     000
```

```
 7591  054054      000      000      000  EF130:  .BYTE   0,0,0,0,0
 7592  054057      000      000
 7593
 7594            054062                    .EVEN
 7595  054062                              BUFFER:
 7596  054062  000402                      BUFONE: .BLKW   258.
 7597  055066  000402                      BUFTWO: .BLKW   258.
 7598            054062                    .       =       BUFFER
 7599                                      .NLIST BEX
       054062                             HELP:
       054062  005015                     .ASCII  <CR><LF>
       054064  044514  052123  047440     .ASCII  @LIST OF TESTS@<CR><LF>
       054103     055  026455  026455     .ASCII  @-------------@<CR><LF>
       054122  005015                     .ASCII  <CR><LF>
       054124  030524  052011  040522     .ASCII  @T1     TRANSFER TEST@<CR><LF>
       054146  031124  041411  047524     .ASCII  @T2     CTOD TEST@<CR><LF>
       054164  031524  046411  051501     .ASCII  @T3     MASSBUS INITIALIZE TEST@<CR><LF>
       054220  032124  041411  042514     .ASCII  @T4     CLEAR STUCK ACTIVE TEST@<CR><LF>
       054254  032524  052011  044522     .ASCII  @T5     TRISTATE TRANSFER TEST@<CR><LF>
       054307     124  004466  042522     .ASCII  @T6     REGISTER SELECT TEST@<CR><LF>
       054340  033524  042011  044522     .ASCII  @T7     DRIVE TYPE TEST@<CR><LF>
       054364  030524  004460  042504     .ASCII  @T10    DEVICE AVAILABLE TEST@<CR><LF>
       054417     124  030461  051411     .ASCII  @T11    SEARCH TIMEOUT TEST@<CR><LF>
       054450  030524  004462  042523     .ASCII  @T12    SET DTE TEST@<CR><LF>
       054472  030524  004463  047506     .ASCII  @T13    FORMAT CHANGE TEST@<CR><LF>
       054522  030524  004464  051120     .ASCII  @T14    PROM STROBE TEST@<CR><LF>
       054550  030524  004465  054523     .ASCII  @T15    SYNC WORD COUNT INHIBIT TEST@<CR><LF>
       054612  030524  004466  054523     .ASCII  @T16    SYNC DETECTION TEST@<CR><LF>
       054643     124  033461  040411     .ASCII  @T17    ABORT SYNC DETECTION TEST@<CR><LF>
       054702  031124  004460  054523     .ASCII  @T20    SYNC GENERATION TEST@<CR><LF>
       054734  031124  004461  051127     .ASCII  @T21    WRITE HEADER TEST@<CR><LF>
       054763     124  031062  044011     .ASCII  @T22    HEADER COMPARE TEST @<CR><LF>
       055015     124  031462  042411     .ASCII  @T23    ECC GENERATION TEST@<CR><LF>
       055046  031124  004464  041505     .ASCII  @T24    ECC DETECTION TEST@<CR><LF>
       055076  005015                     .ASCII  <CR><LF>
       055100  005015                     .ASCII  <CR><LF>
       055102  053523  052111  044103     .ASCII  @SWITCH                 USE@<CR><LF>
       055120  026455  026455  026455     .ASCII  @------         ------------------@<CR><LF>
       055156  020040  032461  004411     .ASCII  @  15           HALT ON ERROR@<CR><LF>
       055203     040  030440  004464     .ASCII  @  14           LOOP ON TEST@<CR><LF>
       055227     040  030440  004463     .ASCII  @  13           INHIBIT ERROR TYPEOUTS@<CR><LF>
       055265     040  030440  004461     .ASCII  @  11           INHIBIT ITERATIONS@<CR><LF>
       055317     040  030440  004460     .ASCII  @  10           BELL ON ERROR@<CR><LF>
       055344  020040  034440  004411     .ASCII  @   9           LOOP ON ERROR@<CR><LF>
       055371     040  020040  004470     .ASCII  @   8           LOOP ON TEST IN SWR<7:0>@<CR><LF>
       055431     040  020040  004467     .ASCII  @   7           TN128@<CR><LF>
       055446  020040  033040  004411     .ASCII  @   6           TN64@<CR><LF>
       055462  020040  032440  004411     .ASCII  @   5           TN32@<CR><LF>
       055476  020040  032040  004411     .ASCII  @   4           TN16@<CR><LF>
       055512  020040  031440  004411     .ASCII  @   3           TN8@<CR><LF>
       055525     040  020040  004462     .ASCII  @   2           TN4@<CR><LF>
       055540  020040  030440  004411     .ASCII  @   1           TN2@<CR><LF>
       055553     040  020040  004460     .ASCII  @   0           TN1@<CR><LF>
       055566  005015     000             .ASCIZ  <CR><LF>
                                          .LIST BEX
 7600            000001                   .END
```

```
ABASE = 176700           1359#   1471    1512
ACDW1 = 000000           1471    1514
ACDW2 = 000000           1471    1515
ACPUOP= 000000           1471    1486
ADDW0 = 000000           1471    1516
ADDW1 = 000000           1471    1517
ADDW10= 000000           1471
ADDW11= 000000           1471
ADDW12= 000000           1471
ADDW13= 000000           1471
ADDW14= 000000           1471
ADDW15= 000000           1471
ADDW2 = 000000           1471    1518
ADDW3 = 000000           1471    1519
ADDW4 = 000000           1471    1520
ADDW5 = 000000           1471    1521
ADDW6 = 000000           1471    1522
ADDW7 = 000000           1471    1523
ADDW8 = 000000           1471
ADDW9 = 000000           1471
ADEVCT= 000000           1471    1477
ADEVM = 000000           1471    1513
AENV  = 000000           1471    1482
AENVM = 000000           1471    1483
AFATAL= 000000           1471    1474
AMADR1= 000000           1471    1499
AMADR2= 000000           1471    1503
AMADR3= 000000           1471    1506
AMADR4= 000000           1471    1509
AMAMS1= 000000           1471    1493
AMAMS2= 000000           1471    1501
AMAMS3= 000000           1471    1504
AMAMS4= 000000           1471    1507
AMSGAD= 000000           1471    1479
AMSGLG= 000000           1471    1480
AMSGTY= 000000           1471    1473
AMTYP1= 000000           1471    1494
AMTYP2= 000000           1471    1502
AMTYP3= 000000           1471    1505
AMTYP4= 000000           1471    1508
AOE   = 001000           1135#   1146    6737    6738    6741    6742    6745    674o
APASS = 000000           1471    1476
APE   = 100000           1338#
APRIOR= 000000           1471
APTCSU= 000040           6066    6707#
APTENV= 000001           6059    6241    6663    6705#
APTSIZ= 000200           2247    6704#
APTSPO= 000100           6061    6665    6706#
ASWREG= 000000           1471    1484
ATA   = 100000           1115#   6718    6719    6720    6723    6724    6727    6728    6729    6730    6731    6732    6733
                         6734    6735    6736    6739    6740    6743    6744    6747    6748
ATESTN= 000000           1471    1475
ATNMSK= 000377           1152#
ATNTBL  031700           2444    2462    6750#
AUNIT = 000000           1471    1473
AUSWR = 000000           1471    1485
```

```
AVECT1= 120254          1360#   1471    1510
AVECT2= 000000          1471    1511
A16    = 000400         1309#
A17    = 001000         1308#
BAI    = 000010         1328#
BB00   = 000001         1245#   4003    4017    4023    4196    4200
BB01   = 000002         1244#   4422    4425    4803    4805
BB02   = 000004         1243#
BB03   = 000010         1242#
BB04   = 000020         1241#
BB05   = 000040         1240#
BB06   = 000100         1239#
BB07   = 000200         1238#
BB08   = 000400         1237#
BB09   = 001000         1236#
BIT0   = 000001         1031#   2291    2829    3258    3260    4090    4227    4280    4779
BIT00  = 000001         1021#   1031    1057    1106    1125    1144    1182    1202    1245    1332    1348
BIT01  = 000002         1020#   1030    1056    1105    1143    1181    1201    1244    1331    1347
BIT02  = 000004         1019#   1029    1055    1104    1142    1180    1200    1243    1330    1346
BIT03  = 000010         1018#   1028    1054    1103    1141    1179    1199    1242    1257    1328    1345
BIT04  = 000020         1017#   1027    1053    1102    1140    1198    1241    1327
BIT05  = 000040         1016#   1026    1052    1139    1178    1197    1240    1326
BIT06  = 000100         1015#   1025    1124    1138    1160    1177    1196    1239    1311    1325    1344
BIT07  = 000200         1014#   1024    1123    1137    1159    1176    1195    1219    1238    1256    1310    1324
BIT08  = 000400         1013#   1023    1101    1122    1136    1158    1175    1194    1237    1309    1322    6145
BIT09  = 001000         1012#   1022    1100    1121    1135    1157    1174    1193    1236    1308    1321    6163    6252
BIT1   = 000002         1030#   2886
BIT10  = 002000         1011#   1099    1120    1134    1156    1173    1192    1218    1235    1255    1307    1320    1343
                        6229
BIT11  = 004000         1010#   1051    1119    1133    1172    1191    1209    1217    1234    1254    1319    1342    6170
BIT12  = 010000         1009#   1118    1132    1171    1190    1216    1233    1253    1318    1341
BIT13  = 020000         1008#   1117    1131    1170    1189    1208    1232    1252    1305    1317    1340    6236
BIT14  = 040000         1007#   1116    1130    1169    1188    1207    1231    1251    1263    1304    1316    1339    6131
BIT15  = 100000         1006#   1115    1129    1168    1187    1206    1230    1250    1262    1303    1315    1338
BIT2   = 000004         1029#
BIT3   = 000010         1028#
BIT4   = 000020         1027#
BIT5   = 000040         1026#
BIT6   = 000100         1025#
BIT7   = 000200         1024#   2286
BIT8   = 000400         1023#
BIT9   = 001000         1022#
BOTADR  022422          5230*   5248*   5251    5266    5311#
BOTFLG  022424          5216*   5258*   5261    5264*   5312#
BPTVEC= 000014          1038#
BSE    = 100000         1250#
BUFFER  054062          3316    3405    3483    3660    3798    3910    3978    4127    4322    4586    4588    4615    7570
                        7595#   7598
BUFONE  054062          4882    5043    7596#
BUFTWO  055066          7597#
CC     = 004000         1234#
CH     = 002000         1235#
CHRCNT  022425          5217    5235*   5241*   5242    5245*   5249    5254*   5265*   5313#
CKSWR  = 104410         6130    6225    6251    6602#
CLKSNC  024476          4646    4906    4957    5766#
CLOCK   001526          1605#   3328    3348    5337*   5350*   5361*   5602    5678
```

```
CLR    = 000040          1326#  2293   2541   2562   2624   2677   2687   2725   2767   2832   2893   3015   3038
                         3070   3102   3126   3158   3181   3203   3286   3380   3471   3568   3663   4121   4328
                         4609   4876   5433   5508
CLSPRN  030716           2364   2380   2401   6711#
CMNSTA  004254           2316   2451#
CNSL00  031062           2326   6711#
CNSL01  031121           2361   6711#
CNSL02  031146           2370   6711#
CNSL03  031227           2375   6711#
CNSL04  031257           2386   6711#
CNSL05  031333           2396   6711#
CNSL06  031367           2407   6711#
CNSL07  031414           2420   6711#
CONT   = 000100          1196#
CR     = 000015           946#  2434   5233   6105   6115   6711   7562   7599
CRLF   = 000200           947#  2275   6076   6115
CYLMSK= 001777           1224#  2632
DBCK   = 100000          1168#  5625   5642
DBEN   = 040000          1169#  1183   3346   3418   3420   3433   3435   3437   3440   3442   3496   3498   3500
                         3502   3518   3521   3523   3525   3528   3530   3685   3687   3711   3713   3728   3730
                         3750   3752   3762   3764   5592   5625   5627   5634   5636   5642   5644   5673   5675
                         5685   5687   5690   5692   5694   5696   5711   5713   5722   5724
DBL    = 002000          1343#
DCK    = 100000          1129#  1146   5031
DDISP  = 177570           953#  1443   2235
DEBL   = 020000          1170#
DISPLA  001156           1443#  2235*  2243*  6185*  6228*
DISPRE  000174           1370#  2243
DLT    = 100000          1315#
DMD    = 000001          1182#  1183   1202#  2733   2745   3346   3391   3393   3418   3420   3433   3435   3437
                         3440   3442   3476   3478   3496   3498   3500   3502   3518   3521   3523   3525   3528
                         3530   3574   3578   3580   3593   3595   3610   3612   3627   3629   3685   3687   3711
                         3713   3728   3730   3750   3752   3762   3764   5436   5438   5471   5511   5517   5519
                         5535   5537   5555   5557   5592   5625   5627   5634   5636   5642   5644   5673   5675
                         5685   5687   5690   5692   5694   5696   5711   5713   5722   5724
DPE    = 000010          1257#
DPEHI  = 040000          1339#
DPELO  = 020000          1340#
DPR    = 000400          1122#
DRQ    = 004000          1209#
DRVCLR= 000010           1066#
DRY    = 000200          1123#
DSWR   = 177570           952#  1442   2234
DTE    = 010000          1132#  1146   3384   3396   3426   3445   3447   3506   3533   3537
DTO    = 010000          1171#  1183   3418   3420   3433   3435   3437   3440   3442   3496   3498   3500   3502
                         3518   3521   3523   3525   3528   3530   3685   3687   3711   3713   3728   3730   3750
                         3752   3762   3764   5642   5644   5673   5675   5685   5687   5690   5692   5694   5696
                         5711   5713   5722   5724
DULPRT= 024024           1212#  3256   3260   3263
DVA    = 004000          1051#  2299   3290   3292
DVC    = 000200          1256#  2743
EBL    = 020000          1189#
ECH    = 000100          1138#  1146   5031   6737   6738   6745   6746
ECI    = 004000          1217#  2634
ECRC   = 001000          1193#  4254   4265   4667
EDT1    037724           1632   1753   1761   1769   1777   1785   1793   1809   1817   1825   1833   1841   1849
```

		1857	1865	1881	1889	1897	1921	1929	1945	1953	1969	1977	1985	1993
		2009	2025	2033	2041	2049	2135	2143	2167	2175	2183	2191	753?#	
EDT115	037750	1873	2127	2151	7545#									
EDT130	037752	7546#												
EDT132	037754	7547#												
EDT2	037726	1640	7536#											
EDT220	037756	7548#												
EDT47	037732	1937	7538#											
EDT5	037730	1664	1672	1681	1913	7537#								
EDT52	037734	1961	7539#											
EDT57	037736	2001	7540#											
EDT61	037740	2017	7541#											
EDT65	037742	7542#												
EDT71	037744	7543#												
EDT74	037746	7544#												
ED1	053700	7535	7564#											
ED115	054010	7545	7547	7580#										
ED130	054022	7546	7582#											
ED2	053710	7536	7544	7566#										
ED220	054036	7548	7584#											
ED47	053722	7538	7568#											
ED5	053714	7537	7567#											
ED52	053732	7539	7570#											
ED57	053744	7540	7572#											
ED61	053756	7541	7574#											
ED65	053770	7542	7576#											
ED71	054000	7543	7578#											
EECC =	000020	1198#	4667	4729	4752	4753	4763							
EFT1	037760	1633	1754	1762	1770	1778	1786	1794	1810	1818	1826	1834	1842	1850
		1858	1866	1882	1890	1898	1922	1930	1938	1946	1954	1970	1978	1986
		1994	2010	2026	2034	2042	2050	2136	2144	2168	2176	2184	2192	7550#
EFT115	037776	1874	2128	2152	7557#									
EFT130	040000	7558#												
EFT132	040002	7559#												
EFT2	037762	1641	7551#											
EFT220	040004	7560#												
EFT5	037764	1665	1673	1682	1914	7552#								
EFT57	037766	1962	2002	2018	7553#									
EFT65	037770	7554#												
EFT71	037772	7555#												
EFT74	037774	7556#												
EF1	054042	7550	7554	7555	7586#									
EF130	054054	7558	7591#											
EF2	054045	7551	7556	7587#										
EF5	054046	7552	7560	7588#										
EF57	054050	7553	7557	7559	7589#									
EHT1	037614	1631	1752	1760	1768	1776	1784	1792	1808	1816	1824	1832	1840	1848
		1856	1864	1880	1888	1896	1920	1928	1944	1952	1968	1976	1984	1992
		2008	2024	2032	2040	2048	2134	2142	2166	2174	2182	2190	7516#	
EHT115	037670	1872	7527#											
EHT130	037674	7528#												
EHT132	037700	7529#												
EHT145	037704	7530#												
EHT150	037710	2126	2150	7531#										
EHT2	037620	1639	7517#											
EHT213	037714	7532#												

```
EHT220  037720          7533#
EHT47   037634          1936    7520#
EHT5    037624          1663    1671    7518#
EHT52   037640          1960    7521#
EHT57   037644          2000    7522#
EHT61   037650          2016    7523#
EHT65   037654          7524#
EHT7    037630          1680    1912    7519#
EHT71   037660          7525#
EHT74   037664          7526#
EH1     052222          7516    7562#
EH115   053012          7527    7562#
EH130   053110          7528    7562#
EH132   053227          7529    7562#
EH145   053326          7530    7562#
EH150   053444          7531    7562#
EH2     052301          7517    7562#
EH213   053543          7532    7562#
EH220   053641          7533    7562#
EH3     052310          7526    7562#
EH47    052375          7520    7562#
EH5     052317          7518    7562#
EH52    052423          7521    7562#
EH57    052522          7522    7562#
EH61    052621          7523    7562#
EH65    052657          7524    7562#
EH7     052356          7519    7562#
EH71    052734          7525    7562#
EMS1    040006          6761    6762    7562#
EMS10   040411          6933    6938    7562#
EMS100  045051          6836    6843    6845    6851    6854    6858    6861    6911    7562#
EMS101  045111          6836    6911    7562#
EMS102  045165          6847    6849    6851    6854    6863    7562#
EMS103  045233          6865    7562#
EMS104  045273          6856    6858    6868    7562#
EMS105  045320          6870    7562#
EMS106  045361          6876    7562#
EMS107  045426          6878    6881    6886    7562#
EMS11   040454          6943    7562#
EMS110  045436          6881    7562#
EMS111  045475          6884    7562#
EMS112  045513          6891    7562#
EMS113  045551          6894    6914    7562#
EMS114  045610          6897    7562#
EMS115  045656          6908    7562#
EMS116  045673          6916    7562#
EMS12   040545          6921    7562#
EMS13   040623          6951    6954    6957    6960    7562#
EMS14   040672          6964    6974    7562#
EMS15   040737          6964    6971    7089    7321    7325    7363    7368    7400    7432    7471    7562#
EMS16   041015          6976    6981    6986    7341    7562#
EMS17   041076          6976    6981    6989    7562#
EMS2    040057          6769    7053    7096    7147    7155    7164    7186    7347    7358    7562#
EMS20   041137          6991    6996    7000    7004    7009    7014    7021    7562#
EMS21   041203          6991    6996    7004    7009    7014    7018    7373    7405    7437    7476    7562#
EMS22   041262          6996    7000    7009    7014    7562#
```

```
EMS23   041327          7023    7028    7036    7562#
EMS24   041405          7023    7028    7033    7391    7423    7455    7507    7562#
EMS25   041464          7038    7043    7051    7562#
EMS250  045706          7070    7074    7078    7082    7210    7219    7253    7262    7300    7562#
EMS251  045744          7273    7292    7562#
EMS252  046010          6828    6840    7114    7118    7352    7562#
EMS253  046043          6828    6873    7114    7118    7562#
EMS254  046076          6828    6960    6971    6986    7018    7033    7048    7065    7562#
EMS255  046137          7086    7100    7210    7223    7239    7253    7266    7562#
EMS256  046201          6889    7210    7215    7562#
EMS257  046231          6900    7562#
EMS26   041542          7038    7043    7048    7378    7410    7442    7502    7562#
EMS260  046265          6943    7562#
EMS261  046316          7093    7562#
EMS262  046355          7273    7296    7308    7313    7562#
EMS27   041621          7053    7057    7061    7068    7244    7479    7481    7485    7510    7562#
EMS3    040126          7129    7151    7159    7199    7206    7249    7255    7259    7341    7562#
EMS30   041665          7059    7061    7065    7078    7082    7100    7107    7562#
EMS300  046420          6761    6769    6840    6873    6924    6928    6951    7053    7096    7147    7155    7164    7186
                        7210    7215    7219    7223    7347    7358    7562#
EMS301  046436          6762    6766    6773    6777    6780    6828    6943    6960    6971    6986    7018    7033    7048
                        7065    7122    7202    7562#
EMS302  046461          6762    6766    6769    6780    7562#
EMS303  046506          6762    6943    7562#
EMS304  046517          6762    7562#
EMS306  046526          6766    6777    6828    7122    7202    7286    7289    7562#
EMS307  046542          6769    7053    7096    7147    7155    7164    7186    7347    7358    7562#
EMS31   041745          7070    7562#
EMS310  046603          6773    7111    7313    7562#
EMS311  046620          6773    6777    7562#
EMS312  046664          6776    7562#
EMS313  046722          6780    6960    6971    6986    7018    7033    7048    7065    7562#
EMS314  046753          6783    6789    6795    6801    6832    7562#
EMS315  047002          6786    6792    6798    6804    6996    7059    7562#
EMS316  047030          6783    6786    6789    6792    6795    6798    6801    6804    7562#
EMS317  047051          6783    6786    7562#
EMS32   042016          7074    7562#
EMS320  047071          6789    6792    7562#
EMS321  047111          6795    6798    7562#
EMS322  047131          6801    6804    7562#
EMS323  047151          7562#
EMS324  047170          7562#
EMS325  047210          7562#
EMS326  047221          7562#
EMS327  047231          6828    6865    6870    6894    6928    7114    7180    7210    7253    7255    7273    7562#
EMS33   042076          7078    7562#
EMS330  047237          6832    7562#
EMS331  047256          6889    7057    7086    7562#
EMS332  047306          6900    6974    6989    7021    7036    7051    7068    7332    7396    7428    7464    7498    7562#
EMS333  047324          6904    6926    6933    7000    7014    7162    7183    7191    7282    7292    7296    7300    7313
                        7562#
EMS334  047340          6851    6938    7000    7009    7172    7273    7282    7562#
EMS335  047350          6807    6825    6851    6854    6933    7172    7176    7191    7277    7376    7394    7408    7426
                        7440    7458    7562#
EMS336  047373          6810    6849    6858    6861    6921    6938    7368    7378    7400    7410    7432    7442    7471
                        7479    7502    7510    7562#
```

```
EMS337  047412  6833  6836  6957  6969  6981  7004  7009  7028  7043  7061  7107  7114  7133
                7168  7172  7176  7180  7196  7227  7253  7270  7279  7562#
EMS34   042135  7082  7199  7562#
EMS340  047426  6807  6810  6825  6851  6854  6858  6861  6933  6938  7191  7368  7378  7400
                7410  7432  7442  7471  7502  7562#
EMS341  047437  6836  6911  6964  6976  6981  6991  7004  7009  7023  7028  7038  7043  7061
                7078  7082  7100  7107  7114  7212  7215  7219  7223  7227  7231  7235  7244
                7304  7562#
EMS342  047446  6921  6933  6938  6948  7332  7562#
EMS343  047500  6884  6948  7562#
EMS344  047514  6948  7562#
EMS345  047543  6954  6957  7104  7239  7273  7562#
EMS346  047571  6911  6954  6964  6976  6991  7023  7038  7104  7231  7235  7239  7244  7304
                7308  7562#
EMS347  047607  6960  6971  6986  7018  7033  7048  7065  7562#
EMS35   042173  7089  7162  7172  7180  7196  7562#
EMS350  047630  6996  7000  7009  7014  7282  7313  7352  7363  7562#
EMS351  047635  6996  7562#
EMS352  047653  6813  6817  7014  7562#
EMS353  047667  6916  6918  7057  7129  7139  7143  7151  7159  7168  7206  7336  7562#
EMS354  047713  7070  7074  7118  7129  7136  7151  7159  7206  7249  7259  7262  7266  7341
                7562#
EMS355  047746  7078  7082  7100  7562#
EMS356  047770  6821  7321  7325  7329  7468  7562#
EMS357  050010  7089  7317  7562#
EMS36   042235  7096  7122  7133  7136  7139  7143  7191  7317  7376  7387  7394  7408  7419
                7426  7440  7451  7458  7489  7562#
EMS360  050032  7089  7321  7562#
EMS361  050060  7325  7562#
EMS362  050106  7093  7562#
EMS363  050127  7111  7286  7289  7313  7562#
EMS364  050144  7118  7262  7266  7562#
EMS365  050165  6854  7118  7129  7139  7143  7151  7159  7206  7332  7336  7341  7562#
EMS366  050172  7126  7562#
EMS367  050215  7129  7139  7143  7151  7159  7206  7336  7562#
EMS37   042265  7100  7562#
EMS370  050265  6843  6865  6870  6894  7139  7387  7419  7451  7481  7562#
EMS371  050302  6914  7143  7485  7562#
EMS372  050331  6807  6810  7172  7176  7183  7317  7341  7352  7363  7562#
EMS373  050363  7239  7273  7562#
EMS374  050367  7253  7562#
EMS375  050420  7292  7562#
EMS376  050434  7296  7300  7308  7352  7562#
EMS377  050450  6807  6821  6854  7332  7341  7352  7373  7391  7405  7423  7437  7455  7476
                7507  7562#
EMS4    040173  6889  7086  7562#
EMS40   042325  7104  7107  7562#
EMS400  050461  6810  6863  7363  7562#
EMS401  050476  6865  6870  6876  6894  6897  7336  7354  7363  7383  7415  7447  7460  7493
                7512  7562#
EMS402  050513  6813  6817  6821  6861  7373  7391  7405  7423  7437  7455  7476  7507  7562#
EMS403  050522  7373  7391  7405  7423  7437  7455  7476  7507  7562#
EMS404  050563  7378  7410  7442  7489  7502  7562#
EMS405  050602  6881  6891  7368  7378  7383  7387  7396  7400  7410  7415  7419  7428  7432
                7442  7447  7451  7460  7464  7471  7481  7485  7489  7493  7498  7502  7512
                7562#
```

```
EMS406  050612          6813  6817  7368  7400  7432  7471  7562#
EMS407  050625          7468  7562#
EMS41   042403          7111  7114  7118  7172  7176  7183  7329  7332  7562#
EMS410  050645          6833  7562#
EMS411  050654          6825  6833  7562#
EMS412  050672          6845  6847  7562#
EMS413  050706          6856  6878  6881  6891  6908  6916  7562#
EMS414  050722          6868  6886  7562#
EMS415  050742          6856  6858  6861  7562#
EMS416  050762          6858  6886  7562#
EMS417  050774          6868  6876  6878  6897  6908  7562#
EMS42   042451          7126  7129  7562#
EMS420  051013          6878  6908  7562#
EMS421  051026          6884  6886  7562#
EMS422  051043          6918  7562#
EMS43   042533          7136  7562#
EMS44   042604          7147  7151  7562#
EMS45   042645          7155  7159  7562#
EMS46   042722          7164  7168  7562#
EMS47   043004          7176  7180  7186  7206  7562#
EMS5    040224          6904  7286  7562#
EMS50   043044          7191  7199  7562#
EMS500  051111          6764  7562#
EMS501  051251          6764  6767  6771  6774  6778  6781  6784  6787  6790  6793  6796  6799  6802
                        6805  6830  6841  6874  6890  6901  6905  6922  6925  6930  6935  6940  6945
                        6966  6978  6983  6993  6998  7002  7006  7012  7016  7025  7030  7040  7045
                        7091  7098  7112  7116  7120  7123  7134  7137  7161  7169  7174  7178  7182
                        7188  7193  7197  7200  7203  7208  7213  7217  7229  7233  7237  7241  7246
                        7256  7260  7276  7280  7284  7287  7290  7294  7298  7302  7306  7310  7315
                        7319  7323  7327  7330  7334  7338  7344  7355  7360  7365  7562#
EMS502  051276          6764  6767  6771  6774  6778  6781  6784  6787  6790  6793  6796  6799  6802
                        6805  6830  6935  6940  6945  7250  7349  7355  7495  7562#
EMS503  051345          6764  6771  6774  6778  6781  6809  6812  6815  6819  6823  6827  6830  6835
                        6838  6841  6848  6850  6867  6872  6874  6877  6883  6893  6896  6898  6915
                        6952  6955  6958  6962  6966  6970  6973  6978  6983  6988  6993  7006  7020
                        7025  7030  7035  7040  7045  7050  7055  7058  7063  7067  7071  7075  7094
                        7123  7127  7131  7141  7145  7149  7153  7157  7161  7166  7169  7188  7193
                        7250  7310  7338  7344  7349  7355  7360  7365  7371  7375  7381  7385  7389
                        7393  7398  7403  7407  7413  7417  7421  7425  7430  7435  7439  7445  7449
                        7453  7457  7462  7466  7469  7474  7478  7483  7487  7491  7495  7500  7505
                        7509  7514  7562#
EMS504  051371          6774  6778  6781  6815  6819  6823  6838  6844  6846  6853  6857  6860  6864
                        6869  6880  6885  6888  6910  6913  6917  6919  6930  7071  7075  7080  7084
                        7087  7102  7105  7109  7203  7221  7225  7241  7246  7256  7264  7268  7271
                        7562#
EMS505  051415          6767  7562#
EMS506  051470          6901  7562#
EMS507  051525          6905  7562#
EMS51   043113          7202  7235  7562#
EMS510  051615          7250  7562#
EMS511  051670          6764  6767  6771  6774  6778  6781  6784  6787  6790  6793  6796  6799  6802
                        6805  6809  6812  6815  6819  6823  6827  6830  6835  6838  6841  6844  6846
                        6848  6850  6853  6857  6860  6864  6867  6869  6872  6874  6877  6880  6883
                        6885  6888  6890  6893  6896  6898  6901  6905  6910  6913  6915  6917  6919
                        6922  6925  6930  6935  6940  6945  6952  6955  6958  6962  6966  6970  6973
                        6978  6983  6988  6993  6998  7002  7006  7012  7016  7020  7025  7030  7035
```

```
                    7040    7045    7050    7055    7058    7063    7067    7071    7075    7080    7084    7087    7091
                    7094    7098    7102    7105    7109    7112    7116    7120    7123    7127    7131    7134    7137
                    7141    7145    7149    7153    7157    7161    7166    7169    7174    7178    7182    7188    7193
                    7197    7200    7203    7208    7213    7217    7221    7225    7229    7233    7237    7241    7246
                    7250    7256    7260    7264    7268    7271    7276    7280    7284    7287    7290    7294    7298
                    7302    7306    7310    7315    7319    7323    7327    7330    7334    7338    7344    7349    7355
                    7360    7365    7371    7375    7381    7385    7389    7393    7398    7403    7407    7413    7417
                    7421    7425    7430    7435    7439    7445    7449    7453    7457    7462    7466    7469    7474
                    7478    7483    7487    7491    7495    7500    7505    7509    7514    7562#
EMS52   043162      7227    7231    7239    7244    7562#
EMS53   043223      7249    7493    7562#
EMS54   043264      7253    7259    7262    7266    7562#
EMS55   043341      7270    7562#
EMS56   043417      7273    7277    7279    7352    7562#
EMS57   043512      7282    7562#
EMS6    040274      6924    6928    6933    6938    7562#
EMS60   043576      7282    7562#
EMS600  051767      7212    7215    7219    7223    7562#
EMS601  052017      7227    7562#
EMS602  052037      7231    7235    7562#
EMS603  052100      7304    7562#
EMS604  052121      7368    7373    7378    7383    7387    7391    7396    7562#
EMS605  052146      7400    7405    7410    7415    7419    7423    7428    7562#
EMS606  052164      7432    7437    7442    7447    7451    7455    7460    7464    7562#
EMS607  052204      6891    7471    7476    7481    7485    7489    7493    7498    7502    7507    7512    7562#
EMS61   043650      7289    7562#
EMS62   043723      7292    7296    7300    7304    7562#
EMS63   044004      7308    7313    7317    7321    7325    7329    7332    7336    7383    7415    7447    7562#
EMS64   044044      7341    7562#
EMS65   044114      7347    7354    7562#
EMS66   044155      7358    7363    7368    7378    7400    7410    7432    7442    7468    7471    7502    7562#
EMS67   044232      7396    7428    7464    7498    7562#
EMS7    040340      6926    6928    7562#
EMS70   044344      6813    6969    7460    7512    7562#
EMS71   044412      6807    6810    7562#
EMS72   044471      6807    6810    7562#
EMS73   044546      6813    6817    6821    6825    6833    6861    7562#
EMS74   044616      6813    7562#
EMS75   044671      6817    6821    7562#
EMS76   044745      6825    6833    7562#
EMS77   045006      6817    6821    7562#
EMTVEC= 000030      1041#   2218*   2219*   2489*
EMT1    031710      1630    6761#
EMT10   032076      1687    6783#
EMT100  033546      6960#
EMT101  033566      6964#
EMT102  033606      6969#
EMT103  033620      6971#
EMT104  033644      6976#
EMT105  033664      6981#
EMT106  033704      6986#
EMT107  033730      6991#
EMT11   032114      1695    6786#
EMT110  033750      6996#
EMT111  033772      7000#
EMT112  034012      7004#
```

```
EMT113  034032          7009#
EMT114  034056          7014#
EMT115  034100          7018#
EMT116  034124          7023#
EMT117  034144          7028#
EMT12   032132  1703    6789#
EMT120  034164          7033#
EMT121  034210          7038#
EMT122  034230          7043#
EMT123  034250          7048#
EMT124  034274          7053#
EMT125  034312          7057#
EMT126  034332          7061#
EMT127  034350          7065#
EMT13   032150  1711    6792#
EMT130  034374          7070#
EMT131  034412          7074#
EMT132  034430          7078#
EMT133  034450          7082#
EMT134  034470          7086#
EMT135  034504          7089#
EMT136  034522          7093#
EMT137  034534          7096#
EMT14   032166  1719    6795#
EMT140  034552          7100#
EMT141  034572          7104#
EMT142  034606          7107#
EMT143  034624          7111#
EMT144  034640          7114#
EMT145  034662          7118#
EMT146  034704          7122#
EMT147  034722          7126#
EMT15   032204  1727    6798#
EMT150  034734          7129#
EMT151  034756          7133#
EMT152  034770          7136#
EMT153  035004          7139#
EMT154  035024          7143#
EMT155  035044          7147#
EMT156  035062          7151#
EMT157  035104          7155#
EMT16   032222  1735    6801#
EMT160  035122          7159#
EMT161  035152          7164#
EMT162  035170          7168#
EMT163  035206          7172#
EMT164  035230          7176#
EMT165  035252          7180#
EMT166  035276          7186#
EMT167  035316          7191#
EMT17   032240  1743    6804#
EMT170  035340  2125    7196#
EMT171  035352          7199#
EMT172  035366  2141    7202#
EMT173  035404  2149    7206#
EMT174  035426  2157    7210#
```

```
CZRM
CZRM

$LPV
$MAD
$MAD
$MAD
$MAD
$MAI

$MAM
$MAM
$MAM
$MAM
$MBA
$MFL
$MNE
$MSG
$MSG
$MSG
$MSW
$MTY
$MTY
$MTY
$MTY
$MXC
$NUL
$NWT

$OCN
$OMO
$OVE
$PAS
$PAS
$POW
$PSW
$PWR
$PWR
$PWR
$QUE
$RDC
$RDD
$RDL
$RDO
$RDS
$RES
$RTN
$R2A
$SAV
$SAV
$SCO
$SET

$STU
$SVL
```

```
EMT175  035454            2165    7215#
EMT176  035472            2173    7219#
EMT177  035510            2181    7223#
EMT2    031716            1638    6762#
EMT20   032256            1751    6807#
EMT200  035526            2189    7227#
EMT201  035544            7231#
EMT202  035562            7235#
EMT203  035600            7239#
EMT204  035622            7244#
EMT205  035642            7249#
EMT206  035662            7253#
EMT207  035712            7259#
EMT21   032300            1759    6810#
EMT210  035726            7262#
EMT211  035744            7266#
EMT212  035762            7270#
EMT213  035774            7273#
EMT214  036024            7279#
EMT215  036036            7282#
EMT216  036056            7286#
EMT217  036072            7289#
EMT22   032322            1767    6813#
EMT220  036106            7292#
EMT221  036124            7296#
EMT222  036142            7300#
EMT223  036160            7304#
EMT224  036176            7308#
EMT225  036216            7313#
EMT226  036240            7317#
EMT227  036256            7321#
EMT23   032346            1775    6817#
EMT230  036274            7325#
EMT231  036312            7329#
EMT232  036326            7332#
EMT233  036350            7336#
EMT234  036372            7341#
EMT235  036420            7347#
EMT236  036440            7352#
EMT237  036472            7358#
EMT24   032372            1783    6821#
EMT240  036512            7363#
EMT241  036536            7368#
EMT242  036562            7373#
EMT243  036606            7378#
EMT244  036632            7383#
EMT245  036650            7387#
EMT246  036666            7391#
EMT247  036712            7396#
EMT25   032416            1791    6825#
EMT250  036730            7400#
EMT251  036754            7405#
EMT252  037000            7410#
EMT253  037024            7415#
EMT254  037042            7419#
EMT255  037060            7423#
```

```
EMT256  037104          7428#
EMT257  037122          7432#
EMT26   032436          1799    6828#
EMT260  037146          7437#
EMT261  037172          7442#
EMT262  037216          7447#
EMT263  037234          7451#
EMT264  037252          7455#
EMT265  037276          7460#
EMT266  037314          7464#
EMT267  037332          7468#
EMT27   032470          1807    6833#
EMT270  037346          7471#
EMT271  037372          7476#
EMT272  037416          7481#
EMT273  037434          7485#
EMT274  037452          7489#
EMT275  037470          2133    7493#
EMT276  037510          1967    7498#
EMT277  037526          7502#
EMT3    031744          1646    6766#
EMT30   032510          1815    6836#
EMT300  037552          7507#
EMT301  037576          7512#
EMT31   032530          1823    6840#
EMT32   032544          1831    6843#
EMT33   032556          1839    6845#
EMT34   032570          1847    6847#
EMT35   032602          1855    6849#
EMT36   032614          1863    6851#
EMT37   032634          1871    6854#
EMT4    031764          1654    6769#
EMT40   032664          1879    6858#
EMT41   032706          1887    6861#
EMT42   032734          1895    6865#
EMT43   032752          1903    1911    6868#
EMT44   032766          6870#
EMT45   033004          1919    6873#
EMT46   033020          1927    6876#
EMT47   033034          1935    6878#
EMT5    032006          1662    6773#
EMT50   033052          1943    6881#
EMT51   033070          6884#
EMT52   033104          1959    6886#
EMT53   033122          6889#
EMT54   033136          1975    6891#
EMT55   033154          1951    1983    6894#
EMT56   033172          1991    6897#
EMT57   033206          1999    6900#
EMT6    032032          1670    6777#
EMT60   033222          2007    6904#
EMT61   033236          2015    6908#
EMT62   033254          2023    6911#
EMT63   033272          2031    6914#
EMT64   033304          2039    6916#
EMT65   033320          2047    6918#
```

```
EMT66   033332            2054    6921#
EMT67   033346            2061    6924#
EMT7    032054            1679    6780#
EMT70   033364            2068    6928#
EMT71   033404            2075    6933#
EMT72   033430            2082    6938#
EMT73   033454            2089    6943#
EMT74   033474            2096    6948#
EMT75   033504            2103    6951#
EMT76   033516            2110    6954#
EMT77   033532            2117    6957#
ENBSCH  023326            3321    3410    3488    3667    3803    3915    3983    4135    4332    4628    4888    5492#
EQUALS  030717            6711#
ERR     = 040000          1116#
ERRNMB  022420            5197*   5198*   5201    5213    5218    5310#
ERRTYP  021662            5176#   6238
ERRVEC= 000004            1034#   2232    2233*   2244*   5330    5331    5332*   5333*   5348*   5369*   5370*   6136    6137*
                          6139*   6142*
ERTY00  022426            5183    5315#
ERTY01  022434            5191    5316#
ERTY02  022444            5200    5318#
ERTY03  022453            5206    5320#
ERTY04  022461            5302    5322#
ESRC    = 004000          1191#   5649    5653
FER     = 000020          1140#   1146    4577
FMT16   = 010000          1216#   2634    3315    3404    3548    3550    3659    3797    3909    3977    4126    4324    4614
                          4881
FNCDTB  031600            6715#
FNCMSK= 000077            1058#
F0      = 000002          1056#
F1      = 000004          1055#
F2      = 000010          1054#
F3      = 000020          1053#
F4      = 000040          1052#
GNS     = ****** U        1369    2274    5135    5142    6593    6594    6595    6596    6597    6598    6600    6602    6603
                          6604    6605    6606    6607
GO      = 000001          1057#   3318    3407    3485    3662    3800    3912    3980    4132    4325    4619    4885    5444
                          5477    5564
GTSWR   = 104407          2269    2485    6600#
HCE     = 000200          1137#   1146    4577    6737    6738    6741    6745    6746
HCI     = 002000          1218#   2634    4614    4881
HCRC    = 000400          1136#   1146    4577
HELP      054062          2346    7599#
HELPOS  030727            2340    6711#
HT      = 000011           944#   5239    6074    6115
IAE     = 002000          1134#   6719    6729    6737    6738    6741    6742    6745    6746
IDXMSK= 000077            1297#
IE      = 000100          1311#   1344#
ILF     = 000001          1144#   6718    6727    6728    6730    6731    6732    6733    6734    6735    6736    6739    6740
                          6743    6744    6747    6748
ILF02   = 000002          1063#
ILF24   = 000024          1073#
ILF26   = 000026          1074#
ILF30   = 000030          1076#
ILF32   = 000032          1077#
ILF34   = 000034          1078#
```

```
ILF36 = 000036          1079#
ILF40 = 000040          1080#
ILF42 = 000042          1081#
ILF44 = 000044          1082#
ILF46 = 000046          1083#
ILF54 = 000054          1086#
ILF56 = 000056          1087#
ILF64 = 000064          1090#
ILF66 = 000066          1091#
ILF74 = 000074          1094#
ILF76 = 000076          1095#   2628    2649    2681    2697    2771    2808    2844    2869    3109    3116
ILR   = 000002          1143#
ILRG50= 000050          1283#
ILRG52= 000052          1284#
ILRG54= 000054          1285#
ILRG56= 000056          1286#
ILRG60= 000060          1287#
ILRG62= 000062          1288#
ILRG64= 000064          1289#
ILRG66= 000066          1290#
ILRG70= 000070          1291#
ILRG72= 000072          1292#
ILRG74= 000074          1293#
ILRG76= 000076          1294#
IOTVEC= 000020          1039#   2216*   2217*
IPCK0 = 000001          1348#
IPCK1 = 000002          1347#
IPCK2 = 000004          1346#
IPCK3 = 000010          1345#
IR    = 000100          1325#   2549    2570
IVC   = 010000          1253#   6718    6719    6720    6722    6723    6724    6727    6728    6729    6730    6731    6732
                        6733    6734    6735    6736    6737    6738    6739    6740    6741    6742    6743    6744    6745
                        6746    6747    6748
LBC   = 002000          1255#
LBT   = 002000          1120#
LCLOCK  022742          5350    5379#
LCOUNT  023012          5352    5397#
LF    = 000012           945#   5237    6109    6115    6711    7562    7599
LS    = 000004          1200#
LSC   = 004000          1254#
LST   = 000002          1201#
LSTOP   023054          5351    5411#
MCLK  = 004000          1172#   3578    3593    3610    3627    3685    3711    3728    3750    3762    3832    3851    3870
                        3878    3943    4029    4042    4070    4094    4154    4181    4209    4230    4257    4283    4355
                        4381    4407    4433    4455    4480    4491    4499    4514    4522    4537    4563    4757    4782
                        4816    4839    4968    4981    4989    5060    5084    5517    5535    5711    5722    5742    5752
                        5784    5804    5813
MCPE  = 020000          1305#
MDF   = 000100          1177#
MDPE  = 000400          1322#
MI    = 000004          1180#   3393    3478    3518    5555    5673
MOC   = 000400          1175#   1183    3346    3418    3420    3433    3435    3437    3440    3442    3496    3498    3500
                        3502    3518    3521    3523    3525    3528    3530    3685    3687    3711    3713    3728    3730
                        3750    3752    3762    3764    5592    5625    5627    5636    5642    5644    5673    5675    5685
                        5687    5690    5692    5694    5696    5711    5713    5722    5724
MOH   = 020000          1208#
```

```
MOL    = 010000          1118#
MRD    = 002000          1173#  3829   3832   3834   3874   4495   4518   4968   4970   4985   5808
MR1AAA= 051401           1183#  3829   3832   3834   3851   3853   3870   3938   3943   3945   4029   4031   4042
                         4044   4070   4072   4094   4096   4154   4156   4181   4183   4209   4211   4230   4232
                         4257   4259   4283   4285   4355   4357   4381   4383   4407   4409   4433   4435   4455
                         4457   4480   4482   4491   4514   4537   4539   4563   4565   4757   4759   4782   4784
                         4816   4818   4839   4841   4968   4970   4981   5060   5062   5084   5086   5742   5744
                         5752   5754   5784   5786   5804
MS     = 000040          1178#  3418   3435   3440   3498   3523   3528   3938   3943   3945   5685   5692
MSC    = 000002          1181#  3433   3435   3437   3496   3498   3500   3521   3523   3525   5690   5692   5694
MSE    = 100000          1262#
MSER   = 000200          1176#
MUR    = 001000          1174#  1183   3346   3418   3420   3433   3435   3437   3440   3442   3496   3498   3500
                         3502   3518   3521   3523   3525   3528   3530   3685   3687   3711   3713   3728   3730
                         3750   3752   3762   3764   5438   5471   5517   5519   5535   5537   5555   5557   5592
                         5625   5627   5634   5636   5642   5644   5673   5675   5685   5687   5690   5692   5694
                         5696   5711   5713   5722   5724
MWD    = 000010          1199#  4061   4088   4225   4278   4777
MWP    = 000010          1179#
MXF    = 001000          1321#
NDTMSK= 115760           1146#
NED    = 010000          1318#  2296   2545   2560   2566   2579
NEM    = 004000          1319#
NOP    = 000000          1062#
NOTEX    031553         2304   6711#
NSA    = 100000          1206#
OCC    = 100000          1187#
OFD    = 000200          1219#  2634
OFFSET= 000014           1068#  5477
OM     = 000001          1125#  5480   5482
OPE    = 020000          1252#
OPI    = 020000          1131#  3337   3355   3357   6717   6718   6719   6720   6721   6722   6723   6724   6725
                         6726   6727   6728   6729   6730   6731   6732   6733   6734   6735   6736   6737   6738
                         6739   6740   6741   6742   6743   6744   6745   6746   6747   6748
OR     = 000200          1324#
PACACK= 000022           1072#  5444   5454   5564   5574
PAKACK= 000022           1071#  1072
PAR    = 000010          1141#
PAT    = 000020          1327#
PCLOCK  022724           5337   5375#
PCOUNT  023012           5339   5396#
PDA    = 000400          1194#  4667   4706   4715   4729   4730   4733   4752   5004   5008
PGE    = 002000          1320#
PGM    = 001000          1121#
PHA    = 000200          1195#  4167   4172   4239   4243   4667   4802   4804
PIP    = 020000          1117#
PIRQ   = 177772           951#
PIRQVE= 000240           1045#
PLCLK   022750           5377   5381#
PLFS   = 002000          1192#  3696   3736   3738   4467   4470   4549   4555   4921   4927   4941   4946   4949
                         5727   5770   5773   5774
PLSTP   023060           5409   5413#
PROMPT  030721           2421   2430   6711#
PR0    = 000000           968#
PR1    = 000040           969#
PR2    = 000100           970#
```

```
PR3    = 000140            971#   2251    2489
PR4    = 000200            972#
PR5    = 000240            973#   5388
PR6    = 000300            974#   5333    5340    5353    5386
PR7    = 000340            975#   5385
PS     = 177776            948#    949
PSEL   = 002000           1307#
PSTOP    023046           5338    5408#
PSW    = 177776            949#
PWRVEC= 000024            1040#   2222*   2223*   6612*   6613*   6622*   6628*   6640*   6641*
QSTMRK  030725            2335    2440    6711#
RD     = 000070           1092#   3800    3912    4885
RDCHR  = 104411           2327    2341    2351    2354    2422    2431    6465    6603#
RDLIN  = 104412           6535    6604#
RDOCT  = 104413           2365    2381    2402    6605#
RDY    = 000200           1310#
READY    004414           2332    2490#   5110    5158
RECAL  = 000006           1065#
RESREG= 104415            5306    6607#
RESVEC= 000010            1035#
REX    = 010000           1190#
RG     = 040000           1188#   5606    5611
RH     = 000072           1093#   4325
RIP    = 000020           1070#
RLEASE= 000012            1067#
RMAS   = 000016           1271#   2593    3047*
RMASI    001344           1541#
RMASO    001420           1571#
RMBA   = 000004           1353#   2589    5588*
RMBAE  = 000050           1356#
RMBAEI   001376           1554#
RMBAEO   001452           1584#
RMBAI    001332           1536#
RMBAO    001406           1566#   3316*   3405*   3483*   3660*   3798*   3910*   3978*   4128*   4322*   4616*   4883*   5588
RMCS1  = 000000           1267#   1351#   2298    2628*   2639    2681*   2691    2771*   2782*   2795    2844*   2857    3109*
                          3289    5444*   5477*   5564*   5598*
RMCS1I   001326           1534#   2639*   2649*   2691*   2697*   2699    2795*   2808*   2812*   2818    2857*   2869*   2874
RMCS1O   001402           1564#   3318*   3407*   3485*   3662*   3800*   3912*   3980*   4132*   4325*   4619*   4885*   5598
RMCS2  = 000010           1354#   2293*   2294*   2295    2541*   2542*   2544    2551    2560    2562*   2563*   2565    2572
                          2579    2591    2624*   2625*   2677*   2678*   2687*   2688*   2725*   2726*   2767*   2768*   2832*
                          2833*   2893*   2894*   3015*   3016*   3038*   3039*   3070*   3071*   3102*   3103*   3126*   3127*
                          3158*   3159*   3181*   3182*   3203*   3204*   3286*   3287*   3380*   3381*   3471*   3472*   3568*
                          3569*   3663*   3664*   4121*   4122*   4328*   4329*   4609*   4610*   4876*   4877*   5433*   5434*
                          5508*   5509*
RMCS2I   001336           1538#
RMCS2O   001412           1568#
RMCS3  = 000052           1357#
RMCS3I   001400           1555#
RMCS3O   001454           1585#
RMDA   = 000006           1268#   2630*   2641    2773*   2784*   2797    2837*   2846*   2859    2897*   2908    3041*   3053
                          3073*   3085    3161*   3169    3186*   3194    3216*   5580*
RMDAI    001334           1537#   2641*   2651    2797*   2813*   2819    2859*   2875    2908*   2920    3053*   3058    3085*
                          3090    3169*   3172    3194*   3197
RMDAO    001410           1567#   3313*   3402*   3480*   3657*   3795*   3907*   3975*   4124*   4321*   4612*   4879*   5580
RMDB   = 000022           1355#   2595
RMDBI    001350           1543#
```

```
RMDBO   001424          1573#
RMDC  = 000034          1277#  2632*  2643   2790*  2803   2841*  2850*  2863   2901*  2912   3020*  3028   3107*
                        3115   3133*  3145   3188*  3208*  3220   5582*
RMDCI   001362          1548#  2643*  2653*  2803*  2810*  2816*  2822   2863*  2871*  2877   2912*  2930   3028*
                        3031*  3032   3115*  3119*  3120   3145*  3151*  3152   3220*  3225   3226
RMDCO   001436          1578#  3314*  3403*  3481*  3658*  3796*  3908*  3976*  4125*  4320*  4613*  4880*  5582
RMDS  = 000012          1269#  3165*  3212*  5446   5450   5479   5484   5566   5570
RMDSI   001340          1539#
RMDSO   001414          1569#
RMDT  = 000026          1274#  3135*  3253   3265
RMDTI   001354          1545#
RMDTO   001430          1575#
RMEC1 = 000044          1281#  3139*
RMEC1I  001372          1552#
RMEC1O  001446          1582#
RMEC2 = 000046          1282#  2597   3083*
RMEC2I  001374          1553#
RMEC2O  001450          1583#
RMER1 = 000014          1270#  2683*  2693   2729*  2736   2775*  2786*  2799   2852*  2865   2903*  2914   3018*
                        3026   3105*  3113   3129*  3141   3184*  3192   3214*  3334   3336   3354   3377   3383
                        3395   3425   3444   3505   3509   3532   3536   4576   4581   5030   5037   5440*  5473*
                        5560*  5594*
RMER1I  001342          1540#  2693*  2701   2736*  2741*  2799*  2814*  2820   2865*  2878   2914*  2936   3026*
                        3029   3113*  3116*  3117   3141*  3146   3192*  3195
RMER1O  001416          1570#
RMER2 = 000042          1280#  2685*  2695   2731*  2738   2779*  2792*  2805   2854*  2867   2905*  2916   3045*
                        3057   3077*  3089   3137*  3163*  3171   3190*  3210*  3222   5442*  5475*  5562*  5596*
RMER2I  001370          1551#  2695*  2698*  2703   2738*  2743*  2805*  2811*  2817*  2823   2867*  2872*  2879
                        2916*  2940   3057*  3063*  3064   3089*  3095*  3096   3171*  3174*  3175   3222*  3228*
                        3229
RMER2O  001444          1581#
RMHR  = 000036          1278#  3024*  3081*
RMHRI   001364          1549#
RMHRO   001440          1579#  7578
RMLA  = 000020          1272#  3079*
RMLAI   001346          1542#
RMLAO   001422          1572#
RMMR1 = 000024          1273#  2733*  2740   3022*  3346*  3391*  3393*  3418*  3420*  3433*  3435*  3437*  3440*
                        3442*  3476*  3478*  3496*  3498*  3500*  3502*  3518*  3521*  3523*  3525*  3528*  3530*
                        3571   3574*  3578*  3580*  3582   3593*  3595*  3597   3610*  3612*  3614   3627*  3629*
                        3631   3685*  3687*  3690   3700   3711*  3713*  3716   3728*  3730*  3735   3740   3750*
                        3752*  3762*  3764*  3766   3772   3826   3829*  3832*  3834*  3836   3851*  3853*  3855
                        3877*  3880*  3882   3938*  3943*  3945*  3947   3954   4029*  4031*  4033   4042*  4044*
                        4046   4060   4070*  4072*  4086   4094*  4096*  4154*  4156*  4158   4175   4181*  4183*
                        4185   4209*  4211*  4223   4230*  4232*  4242   4253   4257*  4259*  4264   4276   4283*
                        4285*  4355*  4357*  4359   4381*  4383*  4385   4407*  4409*  4411   4433*  4435*  4437
                        4455*  4457*  4459   4472   4480*  4482*  4498*  4501*  4521*  4524*  4537*  4539*  4541
                        4557   4563*  4565*  4567   4666   4673   4705   4718   4728   4735   4751   4757*  4759*
                        4765   4775   4782*  4784*  4816*  4818*  4820   4828   4839*  4841*  4843   4851   4920
                        4930   4940   4951   4968*  4970*  4988*  4991*  5003   5012   5060*  5062*  5064   5072
                        5084*  5086*  5088   5096   5436*  5438*  5471*  5511*  5517*  5519*  5521   5527   5535*
                        5537*  5539   5545   5555*  5557*  5592*  5605   5613   5625*  5627*  5634*  5636*  5642*
                        5644*  5648   5652   5673*  5675*  5685*  5687*  5690*  5692*  5694*  5696*  5711*  5713*
                        5715   5722*  5724*  5726   5738   5742*  5744*  5748   5752*  5754*  5769   5776   5784*
                        5786*  5788   5796   5812*  5815*
RMMR1I  001352          1544#  2740*  2745
```

```
RMMR10  001426          1574#  3870*  3874*  3877   3878*  3880   4491*  4495*  4498   4499*  4501   4514*  4518*
                        4521   4522*  4524
RMMR2 = 000040          1279#  3051*  4002   4007   4016   4022   4194   4199   4346   4370   4420   4679   4686
                        5017   5023
RMMR2I  001366          1550#
RMMR2O  001442          1580#
RMOF  = 000032          1276#  2634*  2645   2777*  2788*  2801   2839*  2848*  2861   2899*  2910   3043*  3055
                        3075*  3087   3131*  3143   3167*  3206*  3218   3474*  5584*
RMOFI   001360          1547#  2645*  2655*  2801*  2809*  2815*  2821   2861*  2870*  2876   2910*  2924   3055*
                        3060*  3061   3087*  3092*  3093   3143*  3148*  3149   3218*  3223*
RMOFO   001434          1577#  3315*  3404*  3482*  3659*  3797*  3909*  3977*  4126*  4324*  4614*  4881*  5584
RMR   = 000004          1142#
RMSN  = 000030          1275#  3049*  3111*
RMSNI   001356          1546#
RMSNO   001432          1576#
RMWC  = 000002          1352#  2587   5586*
RMWCI   001330          1535#
RMWCO   001404          1565#  3317*  3406*  3484*  3661*  3799*  3911*  3979*  4131*  4323*  4618*  4884*  5586
RQA   = 100000          1230#
RQB   = 040000          1231#
RTC   = 000016          1069#
SADMSK= 000037          1111#
SAVREG= 104414          5177   6606#
SA1   = 000001          1106#
SA16  = 000020          1102#
SA2   = 000002          1105#
SA4   = 000004          1104#
SA8   = 000010          1103#
SC    = 100000          1303#
SCTCMP  024206          3673   3809   3921   3990   4141   4338   4634   4894   5668#
SCTMSK= 003700          1162#
SC0   = 000100          1160#
SC1   = 000200          1159#
SC2   = 000400          1158#
SC3   = 001000          1157#
SC4   = 002000          1156#
SEARCH= 000030          1075#
SEEK  = 000004          1064#
SETLFS  024314          3815   3927   4640   4900   5707#
SETOM   023224          5469#
SETVV   023074          5432#
SIZCLK  022464          2472   5329#
SKI   = 040000          1251#  2743
SNGPRT= 020024          1211#  3254   3258   3262
STACK = 001100           939#  2214   2535   2621   2674   2722   2763   2977   3249   3283   3309   3373   3464
                        3565   3653   3791   3903   3971   4118   4314   4606   4873
STANDA  003426          2282   2317#
START   002632          1389   2202#  6306   6389
STKLMT= 177774           950#
STOP    001530         1606#  3331   3351   5338*  5351*  5610   5619   5681
SWR     001154         1442#  2212   2234*  2236   2242*  2249*  2267   2483   5178   6131   6145   6148   6163
                        6170   6229   6236   6248   6252   6309   6348   6403*  6620   6633*
SWREG   000176         1371#  2242   2267   2483   6309   6348   6371
SW0   = 000001          1003#
SW00  = 000001           993#  1003
SW01  = 000002           992#  1002
```

```
SW02   = 000004              991#   1001
SW03   = 000010              990#   1000
SW04   = 000020              989#    999
SW05   = 000040              988#    998
SW06   = 000100              987#    997
SW07   = 000200              986#    996
SW08   = 000400              985#    995
SW09   = 001000              984#    994
SW1    = 000002             1002#
SW10   = 002000              983#
SW11   = 004000              982#
SW12   = 010000              981#
SW13   = 020000              980#   5178
SW14   = 040000              979#
SW15   = 100000              978#
SW2    = 000004             1001#
SW3    = 000010             1000#
SW4    = 000020              999#
SW5    = 000040              998#
SW6    = 000100              997#
SW7    = 000200              996#
SW8    = 000400              995#
SW9    = 001000              994#
TADMSK= 003400             1110#
TAG    = 020000             1232#
TAGADR= 001114             1414#   1421
TAP    = 040000             1207#
TA1    = 000400             1101#
TA2    = 001000             1100#
TA4    = 002000             1099#
TBITVE= 000014             1036#
TIME    001522             1603#   5381*  5398*  5400*
TKVEC  = 000060             1043#   6286*  6287*
TPVEC  = 000064             1044#
TRAPVE= 000034             1042#   2220*  2221*
TRE    = 040000             1304#
TRTVEC= 000014             1037#
TST    = 010000             1233#
TSTNMB  022416              5189*  5190*  5192   5309#
TSTQUE  001456              1593#  2455   2456*  2492   2537   2623   2676   2724   2765   2979   3251   3285   3311
                            3375   3466   3567   3655   3793   3905   3973   4120   4316   4608   4875   5105   5107*
                            5111*
TST1    004430              2526#  6190
TST10   010324              3274#  6197
TST11   010446              3301#  6198
TST12   010746              3365#  6199
TST13   011334              3456#  6200
TST14   011760              3557#  6201
TST15   012344              3645#  6202
TST16   013056              3783#  6203
TST17   013516              3895#  6204
TST2    005004              2613#  6191
TST20   013770              3963#  6205
TST21   014512              4110#  6206
TST22   015472              4306#  6207
TST23   016742              4598#  6208
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
TST24   020166          4865#   6209
TST3    005224          2665#   6192
TST4    005440          2713#   6193
TST5    005626          2755#   6194
TST6    006764          2968#   6195
TST7    010144          3240#   6196
TYPBN = 104406          5299    6598#
TYPDS = 104405          5139    5146    5293    6597#
TYPE  = 104401          2261    2304    2326    2329    2335    2340    2343    2346    2350    2353    2361    2364    2370
                        2375    2380    2386    2396    2401    2407    2420    2421    2424    2430    2433    2440    5133
                        5140    5147    5182    5183    5191    5200    5206    5215    5253    5259    5263    5276    5282
                        5284    5302    5304    5884    5950    6019    6079    6231    6239    6303    6315    6369    6370
                        6373    6384    6394    6405    6424    6472    6478    6483    6487    6492    6493    6495    6498
                        6502    6593#   6642
TYPOC = 104402          2363    5296    6372    6594#
TYPON = 104404          6596#
TYPOS = 104403          2306    2377    2398    5186    5194    5203    5209    6595#
T1      004472          2532    2533    2534#
T10     010366          3280    3281    3282#
T11     010510          3306    3307    3308#
T12     011010          3370    3371    3372#
T13     011376          3461    3462    3463#
T14     012022          3562    3563    3564#
T15     012406          3650    3651    3652#
T16     013120          3788    3789    3790#
T17     013560          3900    3901    3902#
T2      005046          2618    2619    2620#
T20     014032          3968    3969    3970#
T21     014554          4115    4116    4117#
T22     015534          4311    4312    4313#
T23     017004          4603    4604    4605#
T24     020230          4870    4871    4872#
T3      005266          2671    2672    2673#
T4      005502          2719    2720    2721#
T5      005670          2760    2761    2762#
T6      007026          2974    2975    2976#
T7      010206          3246    3247    3248#
UBUSQS  030763          2350    6711#
UNS   = 040000          1130#   2741
UNTMSK= 000007          1335#   2548    2569
UPE   = 020000          1317#
USE   = 040000          1263#
U0    = 000001          1332#
U1    = 000002          1331#
U2    = 000004          1330#
VV    = 000100          1124#   5447    5451    5567    5571
WATCH   001524          1604#   3327*   3329    3347*   3349    5401*   5403*   5601*   5608    5677*   5679
WC    = 000040          1197#   3583    3598    3602    3615    3618    3632    3691    3717    3767    3837    3856    3883
                        3887    3948    3952    4034    4047    4159    4186    4360    4386    4412    4438    4460    4542
                        4568    4821    4826    4844    5065    5070    5089    5522    5529    5540    5546    5716    5739
                        5749    5781    5793
WCD   = 000050          1084#
WCE   = 040000          1316#   6737    6738
WCEHI = 010000          1341#
WCELO = 004000          1342#
WCF   = 000040          1139#   1146
```

```
WCH    = 000052          1085#
WD     = 000060          1088#  3318   3407   3485   3662   4619
WH     = 000062          1089#  3980   4132
WLE    = 004000          1133#  1146   6741   6742
WRL    = 004000          1119#
XNUDC  = 176000          1225#  2653   2810   2850   2871   2931   2934   3031   3119   3151   3225
XNUER2= 001567           1258#  2698   2811   2854   2872   2941   2944   3063   3095   3174   3228
XNUOF  = 161577          1220#  2655   2809   2848   2870   2925   2928   3060   3092   3148   3223
XSIZ     003304          2286#  2427
$APTHD   001100          1401   1407#
$ASTAT= ****** U         6685   6700
$ATYC    030474          6656   6658#
$ATY1    030450          6654#
$ATY3    030456          6064   6655#
$ATY4    030466          6244   6657#
$AUTOB   001150          1439#  2271*  2487*  6362   6519
$BASE    001276          1512#  2292   2362   2372*  2536   2603   2622   2675   2723   2764   2978   3250   3284
                         3310   3374   3465   3566   3654   3792   3904   3972   4119   4315   4607   4874
$BDADR   001136          1434#  2550*  2551*  2571*  2572*  2603*  3264*  3265*  3293*  3335*  3336*  3376*  3377*
                         3508*  3509*  3535*  3536*  3570*  3571*  3699*  3700*  3739*  3740*  3771*  3772*  3825*
                         3826*  3953*  3954*  4006*  4007*  4021*  4022*  4174*  4175*  4198*  4199*  4241*  4242*
                         4263*  4264*  4345*  4346*  4471*  4472*  4556*  4557*  4580*  4581*  4672*  4673*  4685*
                         4686*  4717*  4718*  4734*  4735*  4764*  4765*  4827*  4828*  4850*  4851*  4929*  4930*
                         4950*  4951*  5011*  5012*  5022*  5023*  5036*  5037*  5053*  5071*  5072*  5095*  5096*
                         5449*  5450*  5483*  5484*  5526*  5527*  5544*  5545*  5569*  5570*  5612*  5613*  5651*
                         5652*  5775*  5776*  5795*  5796*  7564   7566   7572   7580   7582   7584
$BDDAT   001142          1436#  2544*  2545   2565*  2566   2826*  2883*  2954*  3253*  3254   3256   3258   3260
                         3289*  3290*  3334*  3337*  3354*  3355*  3383*  3384*  3395*  3396*  3425*  3426*  3444*
                         3445*  3447*  3505*  3506*  3532*  3533*  3582*  3583*  3597*  3598*  3614*  3615*  3631*
                         3632*  3695*  3696*  3735*  3736*  3770*  3836*  3837*  3855*  3856*  3882*  3883*  3947*
                         3948*  4002*  4003*  4016*  4017   4101*  4102   4173*  4194*  4195*  4196   4266*  4373*
                         4424*  4469*  4554*  4576*  4577*  4666*  4667*  4669   4679*  4680*  4682   4716*  4732*
                         4762*  4825*  4848*  4926*  4927*  4947*  4949*  5009*  5020*  5035*  5051*  5069*  5093*
                         5446*  5447*  5479*  5480*  5528*  5546*  5566*  5567*  5605*  5606*  5648*  5649*  5772*
                         5773*  5792*  5793*  7564   7567   7572   7576   7578   7580   7582   7584
$BELL    001212          1459#  6231   6264   6315   6512
$BIN     025104          5880*  5883*  5884   5891#
$CDW1    001302          1514#
$CDW2    001304          1515#
$CHARC   026036          6081*  6091*  6098   6107*  6112#
$CKSWR   027122          6348#  6602
$CMTAG   001114          1422#  2209   2210   2218   2224   2225   2226
$CM3   = 000000          1452#
$CM4   = 000005          1452#  1453#  1454#  1455#  1456#  1457#
$CNTLC   030020          6303   6384   6512#
$CNTLG   030032          6369   6514#
$CNTLU   030025          6394   6487   6513#
$CPUOP   001250          1486#
$CRLF    001217          1461#  5147   5182   5215   5253   5263   5276   5284   5304   6080   6115   6239   6264
                         6405   6492   6512
$DBLK    025322          5916   5950   5958#
$DDW0    001306          1516#
$DDW1    001310          1517#
$DDW2    001312          1518#
$DDW3    001314          1519#
$DDW4    001316          1520#
```

CROSS REFERENCE TABLE -- USER SYMBOLS

```
$DDW5   001320          1521#
$DDW6   001322          1522#
$DDW7   001324          1523#
$DEVCT  001232          1477#
$DEVM   001300          1513#   2289*   2301*   2419*   2428*   2446*   2447    2454
$DOAGN  021652          5129    5150    5156#
$DTBL   025312          5919    5954#
$ENDAD  021642          1380    2259    5152#   6259
$ENDCT  021506          2224    5131#
$ENULL  021656          5159#
$ENV    001242          1482#   2265    2281    2481    6059    6241    6663    6687
$ENVM   001243          1483#   2247    2286    6061    6066    6665
$EOP    021452          5121#
$EOPCT  021500          2224*   5128#   5132
$EOSP   021416          2605    3267    5103#
$ERFLG  001117          1425#   6120    6159    6161    6167*   6189    6226*   6264
$ERMAX  001131          1431#   2227*   2531*   2617*   2670*   2718*   2759*   2973*   3245*   3279*   3305*   3369*   3460*
                        3561*   3649*   3787*   3899*   3967*   4114*   4310*   4602*   4869*   6161    6184*   6189
$ERROR  026424          2218    6224#
$ERRPC  001132          1432#   5207    6233*   6234*   6235    6264
$ERRTB  001532          1624#   5222
$ERTTL  001126          1429#   5144    5148*   6232*   6264
$ESCAP  001210          1458#   2226*   6183*   6255    6257    6264
$ETABL  001242          1481#
$ETEND  001326          1413    1524#
$FATAL  001224          1474#   6691*
$FFLG   030714          6654*   6657*   6685    6694*   6702#
$FILLC  001172          1450#   6084    6115
$FILLS  001171          1449#   6115
$GDADR  001134          1433#
$GDDAT  001140          1435#   2547*   2548*   2549*   2568*   2569*   2570*   2827*   2884*   2953*   3292*   3339*   3357*
                        3378*   3510*   3537*   3587*   3602*   3618*   3634*   3698*   3738*   3769*   3827*   3887*   3952*
                        4005*   4023*   4100*   4102    4172*   4200*   4243*   4265*   4374*   4425*   4470*   4555*   4579*
                        4668*   4669    4681*   4682    4715*   4733*   4763*   4826*   4849*   4928*   4946*   5010*   5021*
                        5032*   5033    5052*   5070*   5094*   5451*   5482*   5529*   5547*   5571*   5611*   5653*   5774*
                        5794*   7564    7567    7576    7578    7580    7582
$GET42  021632          5149#
$GTSWR  027212          6370#   6600
$HD   = 000001           916     917
$HIBTS  001100          1408#
$HIOCT  030162          6552*   6557#
$ICNT   001120          1426#   2530*   2616*   2669*   2717*   2758*   2972*   3244*   3278*   3304*   3368*   3459*   3560*
                        3648*   3786*   3898*   3966*   4113*   4309*   4601*   4868*   6174*   6175    6177*   6188
$ILLUP  030432          6612    6628    6645#
$INTAG  001151          1440#   6367*   6386    6406    6519
$ITEMB  001130          1430#   5198    6235*   6243    6264
$LF     001220          1462#   6115    6264    6502    6512
$LFLG   030713          6695*   6701#
$LLCSR  001512          1599#   5349    5379*   5411*
$LLVEC  001514          1600#   5341*   5342*   5352*   5353*   5364*   5365*
$LPADR  001122          1427#   2228*   2532*   2618*   2671*   2719*   2760*   2974*   3246*   3280*   3306*   3370*   3461*
                        3562*   3650*   3788*   3900*   3968*   4115*   4311*   4603*   4870*   6156*   6165*   6181*   6186
                        6188
$LPCSB  001504          1596#   5375*
$LPCSR  001502          1595#   5336    5376*   5408*
$LPERR  001124          1428#   2229*   2533*   2619*   2672*   2720*   2761*   2975*   3247*   3281*   3307*   3371*   3462*
```

```
                    3563*  3651*  3789*  3901*  3969*  4116*  4312*  4604*  4871*  6165  6182*  6188  6254
$LPVEC  001506      1597#  5339*  5340*  5354*  5355*  5362*  5363*
$MADR1  001254      1499#
$MADR2  001260      1503#
$MADR3  001264      1506#
$MADR4  001270      1509#
$MAIL   001222      1409   1413   1472#  2246   2265   2481   2527   2614   2666   2714  2756  2969  3241
                    3275   3302   3366   3457   3558   3646   3784   3896   3964   4111  4307  4599  4866
                    6059   6180   6241
$MAMS1  001252      1493#
$MAMS2  001256      1501#
$MAMS3  001262      1504#
$MAMS4  001266      1507#
$MBADR  001102      1409#
$MFLG   030712      6655*  6661   6696*  6700#
$MNEW   030050      6373   6517#
$MSGAD  001236      1479#  6671*  6674
$MSGLG  001240      1480#  6676*
$MSGTY  001222      1473#  6669   6677*  6689   6693*
$MSWR   030037      6370   6515#
$MTYP1  001253      1494#
$MTYP2  001257      1502#
$MTYP3  001263      1505#
$MTYP4  001267      1508#
$MXCNT  026352      6178   6188#
$NULL   001170      1448#  6086   6115
$NWTST= 000001      2522#  2524   2609#  2611   2661#  2663   2709#  2711   2751#  2753  2964#  2966  3236#
                    3238   3270#  3272   3297#  3299   3361#  3363   3452#  3454   3553#  3555  3641#  3643
                    3779#  3781   3891#  3893   3959#  3961   4106#  4108   4302#  4304   4594#  4596  4861#
                    4863
$OCNT   025554      5991*  6020*  6033#
$OMODE  025556      5986*  5990*  5995   5998*  6009*  6035#
$OVER   026336      6132   6157   6166   6176   6185#
$PASS   001230      1476#  2246*  5125*  5126*  5137   5159   6172   6189
$PASTM  001106      1411#
$POWER  030440      6643   6648#
$PSW    001520      1602#  5384*  5414
$PWRDN  030272      2222   6612#  6640
$PWRMG  030426      6643#
$PWRUP  030344      6622   6628#
$QUES   001216      1460#  6115   6264   6424   6495   6512
$RDCHR  027464      6437#  6603
$RDDEC= ****** U    6606
$RDLIN  027554      6460#  6604
$RDOCT  030062      6530#  6605
$RDSZ = 000010      6453#
$RESRE  024774      5856#  6607
$RTNAD  021654      5158#
$R2A  = ****** U    6608
$SAVRE  024736      5840#  6606
$SAVR6  030436      6621*  6629   6630*  6631*  6647#
$SCOPE  026042      2216   6129#
$SETUP= 000137      1362#  2215   2216   2218   2220   2222   2224   2225   2226   2228  2259  2262  2478
                    5123   6130   6225   6251   6259   6307   6312   6313   6343   6519
$STUP = 177777      1362#
$SVLAD  026302      6140   6179#
```

```
$SVPC = 000200          1378#  1383
$SWR  = 167400           905#   916   921   922   923   924   925   926   927  1457  1458  1459  2225
                        2226  2228  2229  2527  2614  2666  2714  2756  2969  3241  3275  3302  3366
                        3457  3558  3646  3784  3896  3964  4111  4307  4599  4866  5118  5124  5151
                        5157  5159  6121  6122  6123  6124  6125  6131  6143  6145  6146  6159  6160
                        6161  6168  6169  6170  6182  6185  6188  6216  6217  6218  6219  6220  6229
                        6236  6248  6252  6264  6644
$SWREG  001244          1484#  2249
$SWRMK= 000000           927   928  6125  6126  6149
$SW08T  026354          6155  6189#
$TESTN  001226          1475#  2527*  2614*  2666*  2714*  2756*  2969*  3241*  3275*  3302*  3366*  3457*  3558*
                        3646*  3784*  3896*  3964*  4111*  4307*  4599*  4866*  5190  6180*
$TIMES  001206          1457#  2225*  5124*  6168*  6175  6178*  6188
$TKB    001162          1445#  6267  6288  6299  6324  6352  6379
$TKCNT  026624          6268#  6283*  6313  6330*  6444  6446*
$TKINT  026634          2319  2491  6283#  6304  6365
$TKQEN= 026633          6272#  6338  6449
$TKQIN  026626          6269#  6284*  6285  6336*  6337*  6338  6340*
$TKQOU  026630          6270#  6285*  6447  6448*  6449  6451*
$TKQSR  026632          6271#  6284  6340  6451
$TKS    001160          1444#  6267  6289*  6320*  6322  6328*  6350  6366*  6376  6388*  6408*
$TKSRV  026704          6286  6299#
$TMP0   001174          1452#  3262*  3511*  3701*  3839*  3858*  4216*  4218  4236  4293  4506  4507*  4527
                        4586  4770  4788  4792  4977*  4979  5454*  5574*  7568  7570  7572  7574  7576
                        7580  7582
$TMP1   001176          1453#  2328*  2329  2330  2333  2342*  2343  2344  2352*  2353  2356  2366*  2368
                        2372  2373*  2374*  2376  2382*  2384  2388  2389*  2390*  2391*  2392*  2393*  2394*
                        2395*  2397  2403*  2405  2410*  2411*  2412*  2413*  2414*  2415  2423*  2424  2425
                        2432*  2433  2434  2436  2438  2442  3263*  3512*  4217*  4295  4508*  4588  4794
                        4978*  4995  7568  7570  7572  7574  7582
$TMP2   001200          1454#  4297  4509*  4796*  7568  7574
$TMP3   001202          1455#  4797*  7574
$TMP4   001204          1456#
$TN   = 000025           916#  2522  2527#  2532  2609  2614#  2618  2661  2666#  2671  2709  2714#  2719
                        2751  2756#  2760  2964  2969#  2974  3236  3241#  3246  3270  3275#  3280  3297
                        3302#  3306  3361  3366#  3370  3452  3457#  3461  3553  3558#  3562  3641  3646#
                        3650  3779  3784#  3788  3891  3896#  3900  3959  3964#  3968  4106  4111#  4115
                        4302  4307#  4311  4594  4599#  4603  4861  4866#  4870  6150  6190
$TPB    001166          1447#  6104*  6115
$TPFLG  001173          1451#  6053  6115
$TPS    001164          1446#  6102  6115
$TRAP   030164          2220  6566#
$TRAP2  030224          6581#  6592
$TRP  = 000016          6585#  6594#  6595#  6596#  6597#  6598#  6599#  6600  6601#  6602  6603#  6604#  6605#
                        6606#  6607#  6608#
$TRPAD  030236          6575  6592#
$TSTM   001104          1410#
$TSTNM  001116          1424#  5123*  6120  6152*  6179*  6180  6185  6189  6228  6264
$TTYIN  030010          6462  6463  6475  6493  6507  6511#
$TYPBN  025032          5877#  6598
$TYPDS  025106          5904#  6597
$TYPE   025560          6053#  6585  6593  6682
$TYPEC  025772          6083  6090  6097  6102#  6103  6410
$TYPEX  026040          6108  6110  6113#
$TYPOC  025356          5989#  6594
 TYPON  025372          5988  5991#  6596
```

```
$TYPOS  025332          5984#   6595
$UNIT   001234          1478#   2492*   5184
$UNITM  001110          1412#
$USWR   001246          1485#
$VECT1  001272          1510#   2373    2388*   2389    2415*
$VECT2  001274          1511#
$XTSTR  026054          6134#
$$GET4= 000000          5151#
$$SW08= 000025          6189#   6190    6191#   6192#   6193#   6194#   6195#   6196#   6197#   6198#   6199#   6200#   6201#
                        6202#   6203#   6204#   6205#   6206#   6207#   6208#   6209#   6210#
$OFILL  025555          5985*   5989*   5999    6034#
$40CAT= ****** U        6131    6238
.       = 055571        1365#   1369#   1378    1379#   1381#   1383#   1388#   1391#   1397    1398#   1400#   1402#   1414
                        1421#   1463    1593#   2213    2228    2229    2275#   5136#   5159    5160#   5958#   6115    6188
                        6189    6264    6267    6271#   6272    6273#   6511#   6512    6519#   6624    6646    6703#   6713#
                        7594#   7596#   7597#   7598#
.$ASTA= ****** U        6655    6658
.$X     = 001100        1397#   1402
```

CLEAR	905#	2541	2562	2624	2677	2687	2725	2767	2832	2892	3015	3038	3070	3102	3126
	3158	3181	3203	3286	3380	3471	3568	3663	4121	4328	4609	4876	5433	5508	
CLKOFF	905#	3331	3351	5610	5618	5681									
CLKON	905#	3327	3347	5601	5677										
CLKSNC	905#	4645	4904	4956											
COMMEN	1046#														
ENBSCH	905#	3320	3409	3487	3666	3802	3914	3982	4134	4331	4627	4886			
ENDCOM	1046#														
ERR	905#	1629	1637	1645	1653	1661	1669	1678	1686	1694	1702	1710	1718	1726	1734
	1742	1750	1758	1766	1774	1782	1790	1798	1806	1814	1822	1830	1838	1846	1854
	1862	1870	1878	1886	1894	1902	1910	1918	1926	1934	1942	1950	1958	1966	1974
	1982	1990	1998	2006	2014	2022	2030	2038	2046	2053	2060	2067	2074	2081	2088
	2095	2102	2109	2116	2124	2132	2140	2148	2156	2164	2172	2180	2188		
ERROR	940#	2474	2552	2573	2604	2658	2706	2747	2828	2885	2955	3034	3066	3098	3122
	3154	3177	3209	3231	3266	3294	3323	3340	3358	3386	3398	3412	3428	3448	3490
	3513	3538	3588	3603	3619	3635	3669	3675	3702	3741	3773	3805	3811	3817	3840
	3859	3888	3917	3923	3929	3955	3985	3992	4008	4024	4066	4078	4104	4137	4143
	4176	4201	4244	4267	4300	4334	4340	4375	4426	4473	4558	4582	4592	4630	4636
	4642	4648	4674	4687	4719	4736	4766	4798	4829	4852	4890	4896	4902	4908	4931
	4952	4959	5013	5024	5038	5054	5073	5097							
ESCAPE	1046#														
GETAS	905#														
GETBA	905#														
GETBAE	905#														
GETCS1	905#	2637	2690	2794	2856	3288									
GETCS2	905#	2543	2564												
GETDA	905#	2640	2796	2858	2907	3052	3084	3168	3193						
GETDB	905#														
GETDC	905#	2642	2802	2862	2911	3027	3114	3144	3219						
GETDS	905#	5445	5478	5565											
GETDT	905#	3252													
GETEC1	905#														
GETEC2	905#														
GETER1	905#	2692	2735	2798	2864	2913	3025	3112	3140	3191	3333	3353	3382	3394	3424
	3443	3504	3531	4575	5029										
GETER2	905#	2694	2737	2804	2866	2915	3056	3088	3170	3221					
GETHR	905#														
GETLA	905#														
GETMR1	905#	2739	3581	3596	3613	3630	3689	3715	3734	3765	3835	3854	3881	3946	4032
	4045	4058	4085	4157	4184	4222	4251	4274	4358	4384	4410	4436	4458	4540	4566
	4665	4704	4727	4749	4773	4819	4842	4918	4939	5002	5063	5087	5520	5538	5603
	5647	5714	5725	5737	5746	5768	5787								
GETMR2	905#	4001	4015	4193	4369	4419	4678	5016							
GETOF	905#	2644	2800	2860	2909	3054	3086	3142	3217						
GETPRI	1046#	5382													
GETSN	905#														
GETSWR	905#	1046#	2262#	2478											
GETWC	905#														
GETX	905#														
MSG	905#	2524	2611	2663	2711	2753	2966	3238	3272	3299	3363	3454	3555	3643	3781
	3893	3961	4108	4304	4596	4863									
MULT	1046#														
NEWTST	1046#	2522	2609	2661	2709	2751	2964	3236	3270	3297	3361	3452	3553	3641	3779
	3891	3959	4106	4302	4594	4861									
POP	1046#	5368	5370	5861	5945	6553	6633	6634	6697	6698					
PUSH	1046#	5330	5331	5841	5904	6532	6614	6620	6658	6660	6681				

PUTAS 905# 3046
PUTBA 905# 5587
PUTBAE 905#
PUTCS1 905# 2627 2680 2770 2781 2843 3108 5443 5476 5563 5597
PUTCS2 905#
PUTDA 905# 2629 2772 2783 2836 2845 2896 3040 3072 3160 3185 3215 5579
PUTDB 905#
PUTDC 905# 2631 2789 2840 2849 2900 3019 3106 3132 3187 3207 5581
PUTDS 905# 3164 3211
PUTDT 905# 3134
PUTEC1 905# 3138
PUTEC2 905# 3082
PUTER1 905# 2682 2728 2774 2785 2851 2902 3017 3104 3128 3183 3213 5439 5472 5559
5593
PUTER2 905# 2684 2730 2778 2791 2853 2904 3044 3076 3136 3162 3189 3209 5441 5474
5561 5595
PUTHR 905# 3023 3080
PUTLA 905# 3078
PUTMR1 905# 2732 3021 3345 3389 3392 3416 3419 3432 3434 3436 3439 3441 3475 3477
3495 3497 3499 3501 3517 3520 3522 3524 3527 3529 3573 3576 3579 3591 3594
3608 3611 3625 3628 3684 3686 3709 3712 3726 3729 3748 3751 3761 3763 3828
3831 3833 3850 3852 3875 3879 3937 3942 3944 4028 4030 4041 4043 4069 4071
4093 4095 4152 4155 4180 4182 4207 4210 4228 4231 4256 4258 4281 4284 4354
4356 4380 4382 4405 4408 4432 4434 4454 4456 4478 4481 4497 4500 4519 4523
4536 4538 4562 4564 4756 4758 4781 4783 4815 4817 4837 4840 4966 4969 4986
4990 5058 5061 5083 5085 5435 5437 5470 5510 5515 5518 5533 5536 5554 5556
5591 5623 5626 5633 5635 5641 5643 5672 5674 5684 5686 5689 5691 5693 5695
5709 5712 5720 5723 5741 5743 5751 5753 5783 5785 5810 5814
PUTMR2 905# 3050
PUTOF 905# 2633 2776 2787 2838 2847 2898 3042 3074 3130 3166 3205 3473 5583
PUTSN 905# 3048 3110
PUTWC 905# 5585
PUTX 905#
REPORT 1046#
RGBFMC 905# 1527 1557
SCOPE 941# 2526 2613 2665 2713 2755 2968 3240 3274 3301 3365 3456 3557 3645 3783
3895 3963 4110 4306 4598 4865
SCTCMP 905# 3672 3808 3920 3989 4140 4337 4633 4892
SETLFS 905# 3814 3926 4639 4898
SETOM 905#
SETPRI 1046# 2251 5388 5414 6440
SETTRA 6585# 6594 6595 6596 6597 6598 6600 6602 6603 6604 6605 6606 6607
SETUP 1046# 2208
SETVV 905#
SKIP 1046#
SLASH 1046#
SPACE 1046#
STARS 1046# 1376 1394 1396 1403 1417 1463 1466 2522 2525 2609 2612 2661 2664 2709
2712 2751 2754 2964 2967 3236 3239 3270 3273 3297 3300 3361 3364 3452 3455
3469 3541 3553 3556 3641 3644 3677 3721 3755 3779 3782 3820 3847 3867 3891
3894 3932 3959 3962 3996 4052 4106 4109 4146 4214 4247 4270 4289 4302 4305
4397 4448 4486 4504 4529 4594 4597 4696 4722 4790 4861 4864 4910 4934 4962
4975 5027 5041 5114 5163 5328 5373 5490 5504 5577 5589 5599 5620 5630 5637
5657 5670 5698 5702 5761 5825 5870 5894 5961 6038 6117 6212 6266 6343 6358
6429 6453 6522 6560 6610 6626 6653
SWRSU 1046# 2230#

```
TAGS      905#   1526
TRMTRP   6585#
TSTSET    905#   2529    2615    2668    2716    2757    2971    3243    3277    3303    3367    3458    3559    3647    3785
         3897    3965    4112    4308    4600    4867
TYPBIN   1046#
TYPDEC   1046#   5137    5144
TYPNAM    905#   1046#   2255
TYPNUM   1046#
TYPOCS   1046#   2376    2397    5184    5192    5201    5207
TYPOCT   1046#   2362    6371
TYPTXT   1046#   5133    5140
$$CMRE   1415#
$$CMTM   1415#   1452    1453    1454    1455    1456
$$ESCA   1046#
$$NEWT   1046#   2522    2609    2661    2709    2751    2964    3236    3270    3297    3361    3452    3553    3641    3779
         3891    3959    4106    4302    4594    4861
$$SET    6585#   6594    6595    6596    6597    6598    6600    6602    6603    6604    6605    6606    6607
$$SETM   2246#
$$SKIP   1046#
.EQUAT    905#    936
.GETPR    905#
.HEADE    905#    906
.SETUP    905#   1362
.SWRHI    905#    917
.SWRLO    905#    928#
.$ACT1    905#   1374
.$APTB    905#   1464#
.$APTH    905#   1392
.$APTY    905#   6651
.$CATC    905#   1363
.$CMTA    905#   1415
.$DIV     905#
.$EOP     905#   5112
.$ERRO    905#   6210
.$ERRT    905#
.$MULT    905#
.$POWE    905#   6608
.$RAND    905#
.$RDDE    905#
.$RDOC    905#   6520
.$READ    905#   6264
.$SAVE    905#   5823
.$SCOP    905#   6115
.$SIZE    905#
.$TRAP    905#   6558
.$TYPB    905#   5868
.$TYPD    905#   5892
.$TYPE    905#   6036
.$TYPO    905#   5959

. ABS.  056072     000

 ERRORS DETECTED:  0
```

```
DSKZ:CZRMKA.BIN,DSKZ:CZRMKA.SEQ/NL:TOC:MC:MD:CND/LI:ME/SOL/DOC/CRF=DSKM:CZRMKA.P11
RUN-TIME:  39 27 4 SECONDS
RUN-TIME RATIO:  936/71=13.0
CORE USED:  32K  (63 PAGES)

DOCUMENT PAGES:  187
```